Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Fachgebiet Neuroinformatik und Kognitive Robotik

Robuste punktwolkenbasierte Detektion von stehenden und hockenden Personen in einer Einkaufsmarktumgebung

Masterarbeit zur Erlangung des akademischen Grades Master of Science

Jonathan Liebner

Betreuer: Benjamin Lewandowski, M.Sc.
Tim Wengefeld, M.Sc.
Verantwortlicher Hochschullehrer:
Prof. Dr. H.-M. Groß, FG Neuroinformatik und Kognitive Robotik

Die Masterarbeit wurde am 22.01.2018 bei der Fakultät für Informatik und Automatisierung der Technischen Universität Ilmenau eingereicht.

Erklärung: „Hiermit versichere ich, dass ich diese Masterarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe. Alle von mir aus anderen Veröffentlichungen übernommenen Passagen sind als solche gekennzeichnet.“

Ilmenau, 22.01.2018 .....

Jonathan Liebner

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Die Erkennung von Personen ist eine wichtige Fähigkeit im Bereich der mobilen Servicerobotik. Ohne diese ist eine natürliche Interaktion zwischen Menschen und Robotern nicht möglich. Daher ist es von großer Bedeutung, dass ein Roboter Menschen selbstständig und verlässlich erkennen kann. Erst aus der erfolgreichen Detektion von Personen kann eine Planung für die nächsten Schritte der Interaktion und Navigation erfolgen. Verschiedene Einsatzgebiete erschaffen immer neue Anforderungen an Serviceroboter. In einer normalen städtischen Umgebung, zum Beispiel in einer Fußgängerzone, sind fast ausschließlich gehende bzw. stehende Personen vorhanden. In einer Einkaufsmarktumgebung hingegen ist das Verhalten der Kunden deutlich abweichender voneinander. Sie nehmen oft hockende oder gebückte Haltungen an, um weitere Informationen bezüglich eines Produktes herausfinden zu können. Des Weiteren greifen sie in Regale hinein, um Produkte zu nehmen und diese in Einkaufswägen zu legen. Zudem entstehen zahlreiche weitere Posen, die atypisch in Hinblick auf die normale Fortbewegung des Menschen sind. All diese Anforderungen werden somit an einen Serviceroboter in einer Einkaufsmarktumgebung gestellt und müssen daher besonders betrachtet werden.

1.2 Bedeutung für Projekte am Fachgebiet für Neuroinformatik und Kognitive Robotik

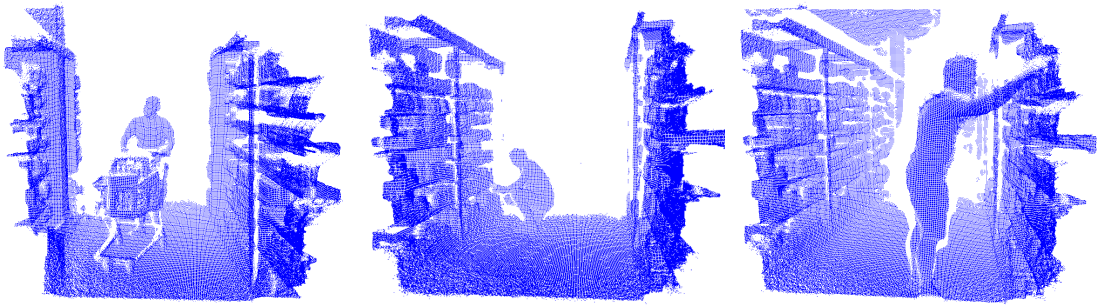
Im Rahmen des Verbundprojektes ROTATOR (D**r**eidimensionale **O**u**t**-of-Stock-Erfassung mittels **a**u**t**onomer m**o**biler **R**oboter) soll ein Verfahren entwickelt werden, mit dem in einem Einkaufsmarkt Out-of-Stocks (OoS) selbstständig von einem mobilen Roboter erfasst werden können. Laut einer Studie von [Gruen et al., 2002] verlieren Einzelhändler ca. vier Prozent des Umsatzes durch Leer- bzw. Fehlbestände im Verkaufsbereich. Besonders kritisch sind dabei spezielle Bereiche, in denen z.B. Werbeaktionen für ausgewählte Artikel stattfinden. Schätzungsweise sind ein Viertel aller OoS sogenannte Shelf-OoS, bei denen die Ware im Lager verfügbar ist und sich lediglich nicht im Verkaufsbereich befindet.
Für einen solchen Roboter ist es von großer Bedeutung, eine sozialverträgliche Navigation sowie eine flüssige Interaktion zu Menschen zu beherrschen. Voraussetzung für diese Fähigkeiten ist eine robuste Personendetektion, um die Umgebungssituation zu kennen. Diese Fähigkeiten sind auch in anderen Einsatzbereichen mobiler Assistenzroboter wie dem häuslichen Umfeld oder einer Klinik verwendbar.

1.3 Ziel der Masterarbeit

Ziel dieser Masterarbeit ist der Entwurf und die Implementierung eines Detektors für stehende und hockende Personen in einer Einkaufsmarktumgebung. Die Detektion findet hierbei in 3D-Punktwolken statt und soll trotz der typischen Voraussetzungen einer Einkaufsmarktumgebung (Verdeckung durch Einkaufswagen, Regale etc.) Personen auch in szenariospezifischen Posen robust erkennen können. Eine Erfassung von Personen sollte bis zu einer Entfernung von 10 Metern möglich sein, hierzu werden verschiedene Ansätze verglichen und evaluiert. Im Rahmen dieser Masterarbeit wird außerdem ein Datensatz für Test- und Trainingszwecke aufgezeichnet. Der Einbezug von Farbe ist eine optionale Erweiterung des zu entwickelnden Verfahrens. Die Ergebnisse und entwickelten Verfahren sollen im Anschluss an diese Masterarbeit auf einer mobilen Roboterplattform in einer Supermarktumgebung einsetzbar sein.

Szenario Einkaufsmarkt

In dem gegebenen Szenario *Einkaufsmarkt* bewegen sich Personen mit zusätzlichen Hilfsmitteln wie Einkaufswagen oder Einkaufskörben. Diese Objekte verdecken Personen zu einem zum Teil sehr großen Grad (siehe Abb. 1.1(a)). Eine weitere häufige Verhaltensweise von einkaufenden Personen ist das Hocken vor Regalen, welche sich die Artikelbeschreibung durchlesen oder sich über die Preise informieren (siehe Abb. 1.1(b)). Für diese Körperhaltung sind bisher bestehende Detektoren für stehende Personen nicht oder nur zum Teil geeignet. Das Hineingreifen in Regale erweitert die Vielfalt der möglichen Körperhaltungen ebenso (siehe Abb. 1.1(c)).



(a) (b) (c)

Abbildung 1.1: ungefärbte Punktwolken typischer Szenen in einem Einkaufsmarkt *(a) Person mit Einkaufswagen, (b) Person hockt vor Regal, (c) Person greift in Regal, Bilder: entommen aus SuPer-Datensatz (**Su**permarkt-**Per**sonen-Datensatz, Erstellung im Rahmen dieser Masterarbeit)*

1.4 Kapitelübersicht

Im folgenden Kapitel 2 werden State-of-the-Art-Verfahren zur Personendetektion vorgestellt. Kapitel 3 beschreibt die theoretischen Grundlagen, die zum Verständnis dieser Masterarbeit notwendig sind. Der eigene Ansatz zur robusten Detektion von stehenden und hockenden Personen in einer Einkaufsmarktumgebung wird in Kapitel 4 vorgestellt und in Kapitel 5 ausgewertet. Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick auf mögliche zukünftige Arbeiten ab.

Kapitel 2

State of the Art

Maschinell Menschen zu erkennen, gewinnt an immer größer werdender Bedeutung. In vielen Einsatzgebieten ist dies bereits erforderlich und wird in Zukunft unvermeidbar sein. In der Automobilbranche dient die Erkennung der Verbesserung der Sicherheit im Straßenverkehr, um Kollisionen mit Personen automatisch zu vermeiden (siehe Abb. 2.1(a)). Erste Paketdienste testen bereits Roboter, die Pakete ausliefern (siehe Abb. 2.1(b)). Auch diese müssen in der Lage sein, Personen zuverlässig zu erkennen. Um dem Roboter eine möglichst gute Entscheidungsgrundlage für die zurückzulegende Strecke und die nächsten Schritte der Navigation zu ermöglichen, ist daher eine genaue Lokalisierung von Menschen in unterschiedlichsten Positionen notwendig.

2.1 Übersicht über mögliche Sensoren zur Personendetektion auf mobilen Robotern

Für die Umgebungswahrnehmung werden verschiedenste Sensoren verwendet, die im Folgenden näher erläutert und verglichen werden. Die Ergebnisse der Klassifikation von Objekten werden durch mehrere Sensoren (z.B. Detektion eines Hindernisses oder einer Person) kombiniert, um einen möglichst robusten, genauen und fehlerfreien Eindruck der Umgebung zu erhalten.

(a) (b)

Abbildung 2.1: Anwendungsszenarios von Personendetektion
(a) Fußgängererkennung im Straßenverkehr, (b) Paketlieferung durch Roboter, Bilder: VOLVO, Starship Technologies

2.1.1 Lasersensoren

Auf vielen Robotersystemen sind 2D-Lasersensoren vorhanden. Laserscanner messen die Zeit, die ein Laserimpuls benötigt, um reflektiert zu werden und zum Sensor zurückzukehren. Als Ergebnis erhält man bei einer typischen Auflösung von 1° die Distanzen für alle Objekte auf einer bestimmten Höhe. Die genauen Messergebnisse der Distanz zu Objekten ermöglicht eine sehr exakte Lokalisierung von Personen. Dies kann sowohl im Innen- als auch im Außenbereich stattfinden, da Lasersensoren Lichtveränderungen gegenüber unempfindlich sind und auch durch Tageslicht keine Verfälschung von Messergebnissen entsteht. Da keine weiteren Informationen außer der Distanz in eine Klassifikation von Objekten einfließen, kann es bei alleinigem Gebrauch zu Fehldetektionen kommen, weshalb Lasersensoren bei der Personendetektion häufig mit weiteren Sensoren kombiniert werden. So verwenden [Wengefeld et al., 2016] Laser- und Kamerasensoren zum Personentracking und visuellen Wiedererkennung von Personen.

2.1.2 Kamerasysteme

Durch die Verwendung von Kameras können viele Informationen über die Umgebung gewonnen werden. Hierbei wird hauptsächlich zwischen 2D- und 3D-Kameras unterschieden. Im Rahmen dieser Masterarbeit wird die Kinect2 verwendet, um eine 3D-

Punktwolke zu erhalten, in der Personen detektiert werden. Eine beispielhafte Punktwolke inklusive einer Detektion von zwei stehenden Personen ist in Abb. 2.2 dargestellt. Im Rahmen dieser Masterarbeit bezeichnet der Begriff 2D-Bild ein RGB-Bild (siehe Abb. 2.2(a)). Das Tiefenbild enthält pro Kamerastrahl im entsprechenden Pixel die metrische Entfernung zum nächstgelegenen Objekt (siehe Abb. 2.2(b)). Eine Punktwolke wird aus der Projektion jedes Pixels des Tiefenbildes in den 3D-Raum erzeugt. Dabei wird der 3D-Punkt aus der Richtung des Kamerastrahls sowie der gemessenen Entfernung bestimmt (siehe Abb. 2.2(c)).

2.2 Verfahren zur Personendetektion in 3D

Es gibt zahlreiche Verfahren für unterschiedliche Sensoren zur Detektion von Personen. Im Rahmen dieser Arbeit werden nur Verfahren betrachtet, die auf der Grundlage von 3D-Daten arbeiten. Einige dieser Verfahren werden im Folgenden näher erläutert. Dabei wird zunächst auf die Möglichkeit eingegangen, Personen durch Laserscanner in 3D zu detektieren sowie im Anschluss Verfahren auf Basis von Tiefenkameras.

2.2.1 Personendetektion in 3D-Laserdaten

Für die Detektion von Personen im dreidimensionalen Raum durch Laserscanner gibt es zwei grundlegende Ansätze. Zum einen können mehrere 2D-Laserscanner verwendet werden, um eine dreidimensionale Erfassung der Umgebung auf verschiedenen Ebenen (Höhe der einzelnen Laserscanner) zu ermöglichen. Zum anderen existieren 3D-Laserscanner, die direkt ein 3D-Modell der Umgebung erzeugen, in dem im Anschluss Personen detektiert werden können. Zwei beispielhafte Scans sind in Abb. 2.3 dargestellt.

Detektion anhand von 3D-Laserscannern

In [Navarro-Sement et al., 2010] wird ein Verfahren gezeigt, das aus einem dreidimensionalen Laserscan Personenhypothesen berechnet. Dabei werden aus dem Laserscan Punktwolken extrahiert, die mögliche Personen darstellen.

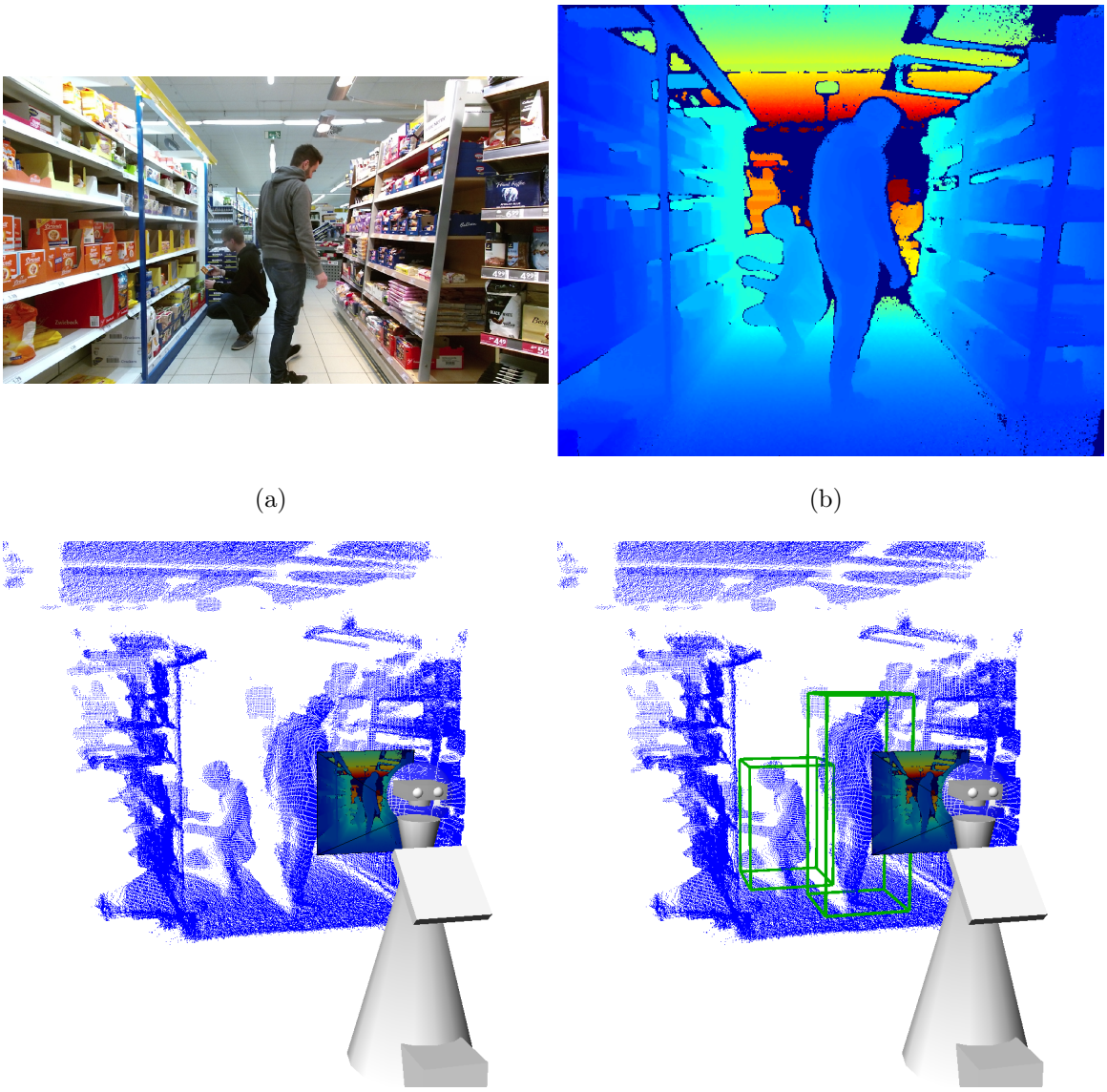
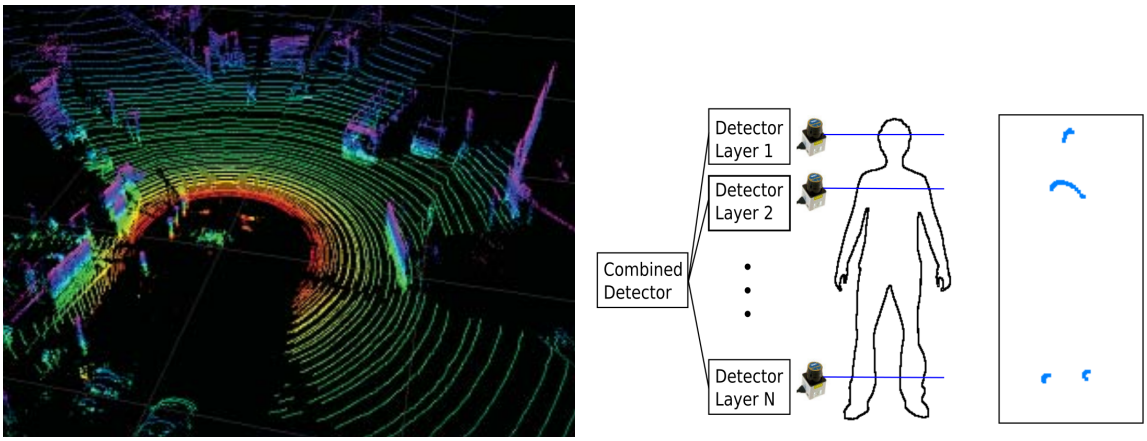


Abbildung 2.2: Aufnahmen aus verschiedenen Kamerasystemen

(a) Aufnahme unter Verwendung einer 2D-RGB-Kamera, (b) Visualisierung der Tiefendaten aus der Kinect als RGB-Bild (blau: kurze Distanz, rot: weite Distanz), (c) Visualisierung der Tiefendaten aus der Kinect2 als Punktwolke, (d) detektierte Personen in der 3D-Punktwolke



(a) (b)

Abbildung 2.3: Zwei Ansätze zur Abbildung eines Raumes durch Laserscans
*(a) Scan eines 3D-Laserscanners entnommen aus [?], (b) Kombination von 2D-Scannern zur Personendetektion aus [*Kidono *et al., 2011]*

Dies geschieht über eine zeitlich integrierte Projektion von 3D-Punkten auf eine virtuelle Grundebene. Aus den entstehenden Segmenten werden nun potentielle Cluster extrahiert. Für jedes Cluster findet nun eine Eigenwertzerlegung durch eine Hauptkomponentenanalyse statt. Entlang der zwei größten Eigenvektoren werden Ebenen aufgespannt, auf welche die Punkte des Clusters in Bins projiziert werden (siehe Abb. 2.4). Durch diese Bins werden letztendlich 2D-Histogramme gebildet, die als Merkmalsvektoren in die Klassifikation durch Support Vektor Maschinen (siehe Kapitel 3.2.2) eingehen. [Navarro-Sement et al., 2010] schließen jedoch aus, dass es überhängende Elemente über Personen wie Dachvorsprünge gibt, da sie ihre Nachforschungen nur in einer freien Außenumgebung durchführen. Diese Fälle sind in einer Supermarktumgebung vorzufinden und müssen bedacht werden. Die Grundidee der Suche nach interessanten Bereichen in einer Projektion auf die Grundebene erscheint für die Verwendung in dieser Masterarbeit sinnvoll.

Detektion anhand mehrerer 2D-Laserscannner

[Carballo et al., 2014] entwarfen ein Verfahren zur Personendetektion anhand von 2D-Laserscans aus unterschiedlichen Höhen (siehe Abb. 2.5). Mehrere Merkmale wie

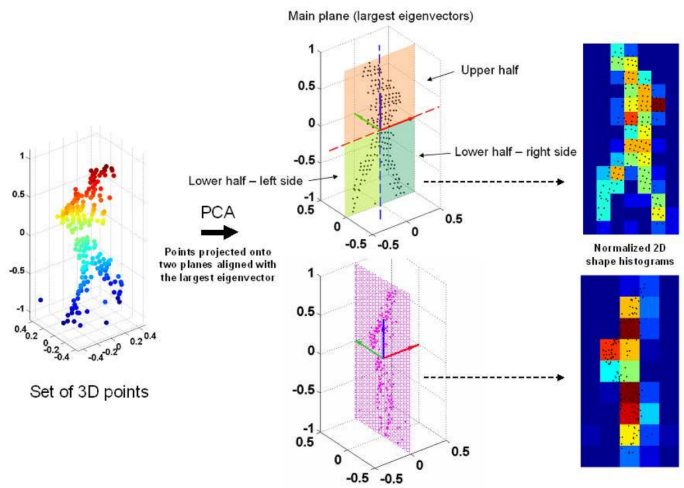
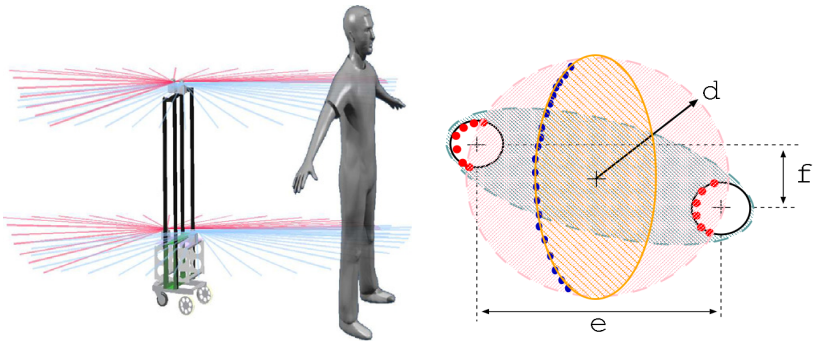


Abbildung 2.4: Generierung von Merkmalsvektoren nach [Navarro-Sement et al., 2010] aus 3D-Laserdaten

links: *Punktwolke eines 3D-Laserscans,* ***mittig:*** *Unterräume des ersten und zweiten Eigenvektors nach der Hauptkomponentenanalyse,* ***rechts:*** *Darstellung der Merkmalsvektoren durch Histogramme, (entnommen aus [Navarro-Sement et al., 2010])*

die Linearität oder die Ähnlichkeit zu einer Ellipse[1] werden dabei aus jeder Ebene extrahiert. In den verschiedenen Ebenen werden die geometrischen Merkmale unterschiedlich bewertet und klassifiziert. Die Bestimmung einer Personenhypothese erfolgt in zwei Schritten. Zunächst werden im Umkreis der den Oberkörper beschreibenden Ellipse die zugehörigen Beine auf einer geringeren Höhe gesucht. Im zweiten Schritt wird bei erfolgreicher Suche die Hypothese anhand des Mittelpunktes der Ellipse definiert. Der Nachteil solcher Verfahren ist durch einen erhöhten Berechnungsaufwand bei der Klassifikation von mehreren Laserscans gegeben, um eine Person zu detektieren. Ein Vorteil des Verfahren ergibt sich jedoch dadurch, dass Personen trotz Verdeckung auf der Beinebene immer noch detektiert werden können. Werden zum Beispiel in einer Wohnraumumgebung die Beine durch einen Tisch verdeckt, könnte die Person weiterhin durch den Laserscan auf Oberkörperhöhe detektiert werden, wenn bei der

[1] Die Ähnlichkeit zu einer Ellipse wurde hier nach der Fitzgibbon-Methode (siehe [Pilu et al., 1996])bestimmt.



(a) (b)

Abbildung 2.5: Detektion von Personen mit Lasern in unterschiedlichen Höhen nach [Carballo et al., 2014]
(a) Darstellung der Positionierung der Laser, (b) Fusionierung zweier Laserebenen; der Oberkörper wird durch die orange Ellipse dargestellt, die zugehörigen Beine entsprechen den zwei Kreisen. d bezeichnet den Ort der gebildeten Personenhypothese, f den Abstand der Beine in der ersten Dimension, e den Abstand der Beine in der zweiten Dimension

Hypothesengenerierung nicht nur die Kombination der Laserebenen verwendet wird, sondern auch die einzelnen Hypothesen der jeweiligen Ebenen.

2.2.2 Personendetektion in Tiefenbildern

In diesem Abschnitt werden unterschiedliche Verfahren vorgestellt, die Tiefenbilder verwenden, um Personen zu detektieren.

Histogramm of Oriented Depth (HOD)

[Spinello und Arras, 2011] entwarfen einen Deskriptor, der sich an die Funktionsweise des HOG-Deskriptors von [Dalal und Triggs, 2005] anlehnt und auf Tiefenbildern statt RGB-Bildern arbeitet. Ein Detektionsfenster fester Größe wird dabei in $n \cdot n$ große Zellen aufgeteilt. Für jede dieser Zellen werden nun die orientierten Tiefengradienten berechnet und in einem 1D-Histogramm gesammelt. Alle Histogramme werden im Anschluss über $2 \cdot 2$ große, sich überlappende Blöcke von Zellen normalisiert. Diese Blockhistogramme werden nun aneinandergehängt und mittels

$$\Delta_x = \frac{D(x+1,y) - D(x-1,y)}{2} \tag{2.1}$$

$$\Delta_y = \frac{D(x,y+1) - D(x,y-1}{2} \tag{2.2}$$

Abbildung 2.6: Berechnung der Tiefendifferenzen in X- und Y-Richtung
$D(x,y)$ entspricht dem Tiefenwert des Pixels (x,y)

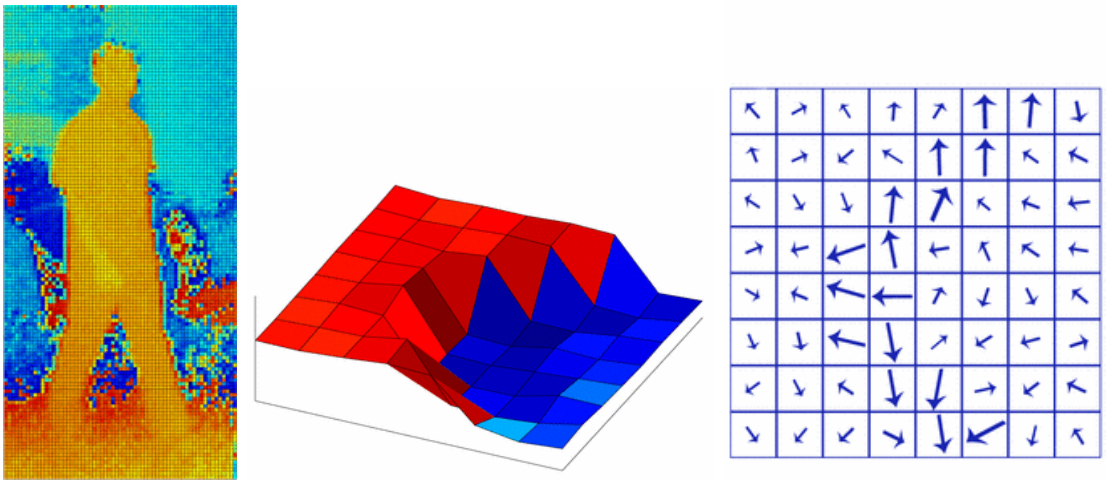
einer SVM eine Trennung zwischen Personen und Nicht-Personen erlernt. Durch die Verwendung von Tiefendaten ergibt sich in der Anwendungsphase ein großer Vorteil zur Beschleunigung des Verfahrens. Um Objekte in unterschiedlicher Entfernung zu klassifizieren, werden bei 2D-Daten häufig Skalenpyramiden aufgebaut und der Detektor auf jede Ebene der Pyramide angewandt. Aus den Tiefeninformationen und einer durchschnittlichen menschlichen Körpergröße leiten [Spinello und Arras, 2011] ab, in welcher Tiefe welche Skalierungsstufen verwendet werden müssen, um eine Person zu detektieren. Hierdurch wird eine deutliche Beschleunigung des Verfahrens erreicht.

Histogram of Oriented Depth (HDD)

Einen ähnlichen Ansatz verfolgen [Wu et al., 2011], die ebenso Histogramme aus Tiefenbildern ableiten. Jedes Detektionsfenster wird wie bei [Dalal und Triggs, 2005] in einzelne Zellen aufgeteilt. Für jeden Pixel (x,y) in einer Zelle findet, wie in Gleichung 2.6, die Berechnung der Tiefendifferenz in X- und Y-Richtung statt. Die zweikomponentige Differenz lässt sich anschaulich als Magnitude und Orientierung darstellen (siehe Abb. 2.7). Pro Zelle wird ein Histogramm der Orientierungen im Intervall [0°,360°) (bei HOG werden die Bins im Intervall [0°,180°) aufgeteilt) gesammelt und durch überlappende Blöcke normalisiert. Die aneinandergereihten Histogramme ergeben wiederum den Featurevektor, der an eine SVM zur Klassifikation übergeben wird.

Simplified Local Ternary Patterns (SLTP)

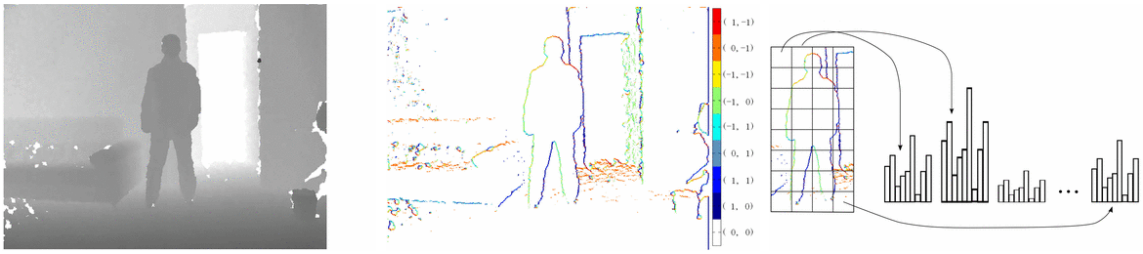
[Yu et al., 2012] verwenden das Prinzip des zuvor vorgestellten HOG-Deskriptors in



(a) (b) (c)

Abbildung 2.7: Ein Tiefenbild und die entsprechenden Tiefendifferenzen nach [Wu et al., 2011]
(a) Person im Tiefenbild, (b) Ausschnitt der Szene im Nackenbereich, (c) Tiefendifferenz der Zellen

Kombination mit Local Ternary Patterns (LTP), die [Tan und Triggs, 2010] zur Gesichtserkennung vorstellten. Für jeden Pixel wird die Tiefendifferenz in der X- und Y-Richtung berechnet (siehe Gleichung 2.3b und 2.3a). $d(x, y)$ entspricht dabei dem Tiefenwert an Bildpixel (x, y). Anhand eines benutzerdefinierten Schwellwertes T_d wird die Tiefendifferenz als -1, 0 oder 1 quantifiziert (siehe Gleichung 2.3c und 2.3d). Dadurch ergeben sich neun Muster, welche die Richtung der Tiefendifferenzen beschreiben. Jedes Detektionsfenster wird in n gleichgroße Blöcke aufgeteilt und aus den berechneten Mustern n Histogramme ermittelt. Da bei gleicher Blockgröße jedes Histogramm die gleiche Anzahl an Elementen besitzt, muss in diesem Verfahren keine Normalisierung durchgeführt werden. Eine beispielhafte Darstellung des SLTP ist in Abbildung 2.8



(a) (b) (c)

Abbildung 2.8: Berechnung des SLTP-Deskriptors anhand eines Beispielbildes *(a) beispielhaftes Tiefenbild einer Szene, (b) aus dem Tiefenbild extrahierte, farbkodierte SLTP-Pattern, (c) Bildung von Histogrammen aus einzelnen Blöcken des Tiefenbildes, Bilder: [Yu et al., 2012]*

dargestellt.

$$\Delta_x = d(x+1, y) - d(x-1, y) \tag{2.3a}$$

$$\Delta_y = d(x, y+1) - d(x, y-1) \tag{2.3b}$$

$$t_x = \begin{cases} 1, & \Delta_x \geq T_d \\ 0, & |\Delta_x| < T_d \\ -1, & \Delta_x \leq -T_d \end{cases} \tag{2.3c}$$

$$t_y = \begin{cases} 1, & \Delta_y \geq T_d \\ 0, & |\Delta_y| < T_d \\ -1, & \Delta_y \leq -T_d \end{cases} \tag{2.3d}$$

$$SLTP(x, y) = (t_x, t_y) \tag{2.3e}$$

Graphbasierte Segmentierung von ROIs

[Choi et al., 2013] nutzen einen anderen Ansatz zur Personendetektion. Anstelle eines simplen *Sliding-Window-Ansatz* findet zunächst eine Segmentierung von Kandidatenregionen statt, dabei wird aus Performanzgründen ein Subsampling des Tiefenbildes durchgeführt. Basierend auf dem Verfahren aus [Felzenszwalb und Huttenlocher, 2004] werden zwei Graphen berechnet. G_{depth} enthält als Knoten

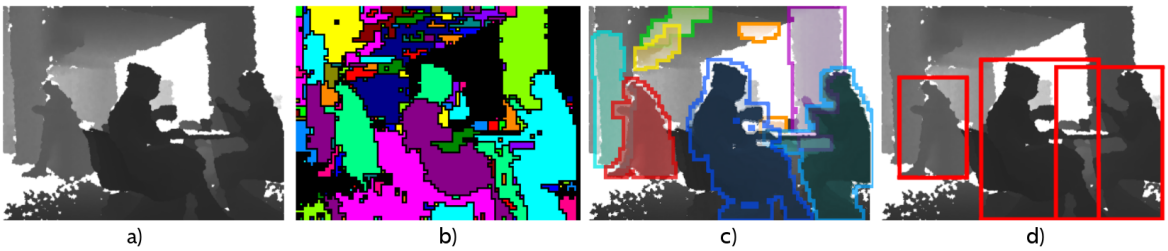


Abbildung 2.9: Personendetektion mittels graphbasierter Segmentierung nach [Choi et al., 2013]
a) Tiefenbild, b) segmentierte Regionen, c) Kandidaten nach Filterung, d) detektierte Personen

alle Punkte des subgesampelten Bildes. Jeder Knoten ist dabei zu den Nachbarn in Haupthimmelsrichtungen (wenn existent) verbunden. Das Gewicht jeder Kante entspricht dabei dem Betrag der Tiefenwertdifferenz der verbundenen Knoten. Auf dem nun entstandenen Graphen werden auf Basis der Tiefenähnlichkeiten Zusammenhangskomponenten gebildet. Für jeden Pixel des Tiefenbildes $p_{i,j}$ wird die Normale $n_{i,j}$ aus dem Punkt selbst sowie den 8 Nachbarn aus derselben Zusammenhangskomponente von G_{depth} berechnet. Hierdurch werden Ausreißer, die im 3D-Raum weit auseinander liegen, entfernt.

Nun wird der zweite Graph G_{normal} analog zu G_{depth} berechnet, bei dem die Kantengewichte der Winkeldifferenz zwischen den entsprechenden Normalen betragen. Das Gewicht einer Kante zwischen zwei Knoten mit Normalen u und v berechnet sich somit durch $\arccos(u \cdot v)$. Wie für G_{depth} werden auch für G_{normal} Zusammenhangskomponenten bestimmt. In einem letzten Schritt werden distinkte Sets von Punkten gebildet, die in G_{depth} und G_{normal} in derselben Zusammenhangskomponente liegen. Im Folgenden werden nur Regionen betrachtet, die eine zu wählende Mindestanzahl an Punkten überschreitet, um Bereiche auszuschließen, die unwahrscheinlich von Personen stammen oder durch Rauschen entstanden sind.
Für diese in Abb. 2.9 b) ersichtlichen ausgewählten Regionen werden nun die Höhe, Breite und durchschnittliche Tiefe in Weltkoordinaten, der Mittelpunkt und der Anteil der auf eine Ebene liegenden Punkte bestimmt. Anhand von aus dem Positivdatensatz

bestimmten Heuristiken werden nun Regionen eliminiert, die keine Personen enthalten können. Die übrig gebliebenen Kandidatenregionen (siehe Abb. 2.9 c))werden nun mittels einer SVM klassifiziert, die als Input den Featurevektor des HOD-Deskriptors (siehe 2.2.2) erhält. Abb. 2.9 d) zeigt die korrekt detektierten Personen in unterschiedlichen Positionen.

Schichtenbasierte Erkennung durch GMMs in Kombination mit CNNs
[Martinson und Yalla, 2016] entwickelten einen kombinierten Ansatz aus der Verwendung von Gaussian Mixture Models (GMMs) sowie Convolutional Neural Networks (CNNs). Zunächst wird eine Segmentierung im Tiefenbild durchgeführt, wodurch einzelne Komponenten entstehen, die Personen enthalten können (siehe Abb. 2.10(a)). Jede Komponente wird nun in Schichten eingeteilt und für jede Reihe eine Gerade und eine Parabel an die Datenpunkte in der X- und Z-Dimension angepasst. Aus den berechneten Parametern der zwei Gleichungen werden im Anschluss sechs geometrische Eigenschaften bestimmt. Hierzu gehören zum Beispiel die Standardabweichung der Datenpunkte zu der Geraden sowie zu der Parabel oder eine Abschätzung der Krümmung der Datenpunkte, welche [Spinello et al., 2010] definierten. Zur Klassifikation werden zwei GMMs bestehend aus je 30 Gaußkurven trainiert, eins für die Positivklasse sowie eins für die Negativklasse. Aus beiden Ergebnissen wird ein Score berechnet, der zur Klassenentscheidung verwendet wird. [Martinson und Yalla, 2016] kombinierten diesen schichtenbasierten Ansatz mit der Verwendung eines *Convolutional Neural Network*, welches ursprünglich von [Krizhevsky et al., 2012] zur Klassifikation der ImageNet-Challenge verwendet wurde. Die letzte Schicht des Netzes ist eine Softmax-Schicht, sodass der Output aus einer Wahrscheinlichkeit besteht, ob es sich bei einem vorliegenden Beispiel um eine Person handelt oder nicht. Diese Wahrscheinlichkeit wird nun gewichtet mit dem Ergebnis der Ausgabe der GMMs zu einer finalen Klassenentscheidung kombiniert. Es wurden zwei CNNs trainiert, eines für die Detektion in 2D-Farbbildern und eines für Tiefenbilder.
Abbildung 2.10(c) zeigt die Precision-Recall-Kurven der einzelnen getesteten Kombinationsmöglichkeiten der GMMs und CNNs. In einer häuslichen Umgebung erreichte bereits der schichtenbasierte Ansatz mit GMMs eine sehr gute Detektionsleistung.

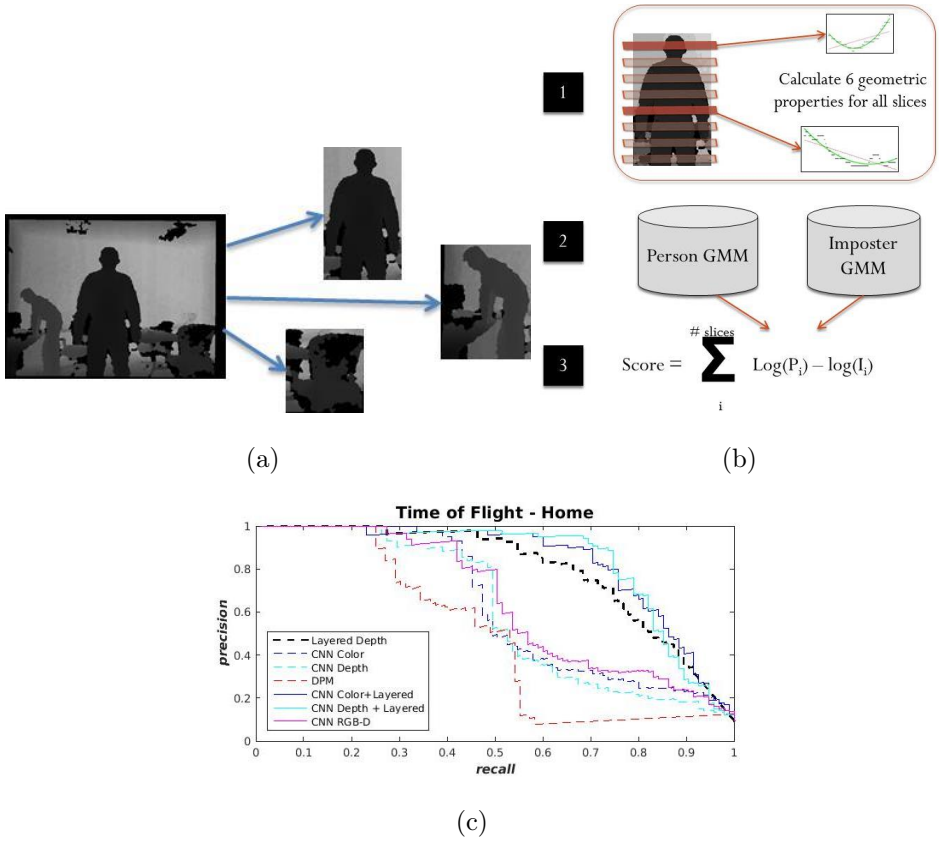


Abbildung 2.10: Personendetektion nach [Martinson und Yalla, 2016] mittels Schichtenbildung im Tiefenbild und Kombination mit einem CNN
(a) segmentierte Komponenten des Tiefenbildes, (b) Scoreberechnung der GMMs, (c) Detektionsergebnis der einzelnen Verfahren

Die Kombination mit einem CNN erzielte noch eine leicht verbesserte Detektion von Personen. Somit erbrachte die Einteilung in Schichten einen großen Vorteil, insbesondere auch bei häufig auftretenden Verdeckungen von Personen in der getesteten häuslichen Umgebung. Eine Verwendung dieses Ansatzes ist somit für das im Rahmen dieser Masterarbeit vorliegende Anwendungsszenario *Einkaufsmarkt* denkbar, da dort auch häufig Verdeckungen von Personen zu erwarten sind.

2.2.3 Personendetektion in Punktwolken

Mit Hilfe der intrinsischen Kameraparameter kann ein Tiefenbild in eine 3D-Punktwolke umgewandelt werden. Für die Detektion von Personen existieren in der bestehenden Literatur nur wenige Ansätze. Mehrere Verfahren wie [Jafari et al., 2014], [Munaro und Menegatti, 2014], [Munaro et al., 2016], [Sun et al., 2016] verwenden 3D-Punktwolken zur Extraktion von Kandidatenclustern, die Personen enthalten könnten. Der Klassifikationsschritt findet jedoch nicht in der 3D-Punktwolke statt, sondern in der entsprechenden Region im 2D-Farbbild oder Tiefenbild. Die Verfahren in diesem Abschnitt verwenden daher nicht ausschließlich 3D-Punktwolken zur Detektion von Personen.

Geometrische Features

[Spinello et al., 2010] haben sich mit der Problematik von teilweise verdeckten Personen beschäftigt. Daher haben sie einen layerbasierten Ansatz gewählt, bei dem stehende Personen in verschiedene Layer eingeteilt werden. Für jedes Layer wird ein eigener AdaBoost-Klassifikator (siehe Abschnitt 3.2.1) mit 20 Weak Learnern trainiert. Über ein euklidisches Clustering werden in jedem Layer Segmente gebildet, für die ein 17-elementiger Featurevektor (siehe Tabelle 2.1) bestimmt wird. Alle Hintergrundsegmente sowie die Cluster der anderen Layer werden für das Training als Negativdaten verwendet. Zusätzlich wird ein Gewicht pro Layer bestimmt, das den Anteil jedes Layers an der Gesamtdetektion angibt.
Zur Laufzeit wird für jedes Segment die Wahrscheinlichkeit durch die AdaBoost-Klassifikatoren bestimmt, dass es zu einem der Layer gehört. Von jedem Segment wird nun ein sogenanntes Votum abgegeben und in einem Voting-Raum zusammengefasst (siehe Abb. 2.11), dabei entspricht das Zentrum einer Person dem lokalen Maximum.

Histogramm von lokalen Oberflächennormalen

Ein histogrammbasiertes Verfahren stellen [Hegger et al., 2013] vor, welches die lokale Verteilung der Oberflächennormalen als Deskriptor nutzt. Es besteht aus 4 Phasen, die jeweils verschiedene Operationen auf den Inputdaten ausführen. Im ersten Schritt wird ein **Preprocessing** (siehe Abb. 2.12 a)) zur Reduzierung der

Nr.	Feature Name	Nr.	Feature Name
f_1	Width	f_{10}	Cubic spline fitting
f_2	Number of points	f_{11}	Standard dev. w.r.t. centroid
f_3	Circularity	f_{12}	Mean avg. dev. from median
f_4	Linearity	f_{13}	Kurtosis w.r.t. centroid
f_5	Boundary length	f_{14}	Radius
f_6	Boundary regularity	f_{15}	PCA ratio
f_7	Mean angular difference	f_{16}	Bounding box area
f_8	Mean curvature	f_{17}	Convex hull area
f_9	Quadratic spline fitting		

Tabelle 2.1: Features nach [Spinello et al., 2010]

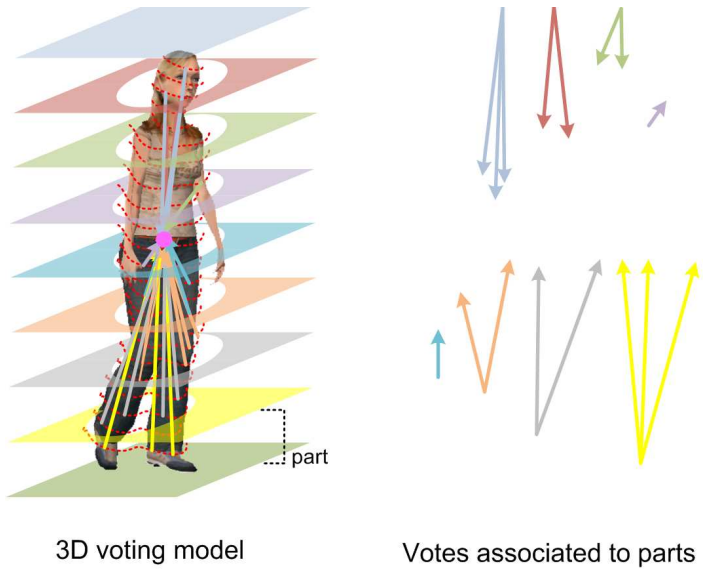


Abbildung 2.11: 3D Voting-Modell nach [Spinello et al., 2010]

links: Aufteilung der Punktwolke in einzelne Layer, rechts: Votes der jeweiligen Layer

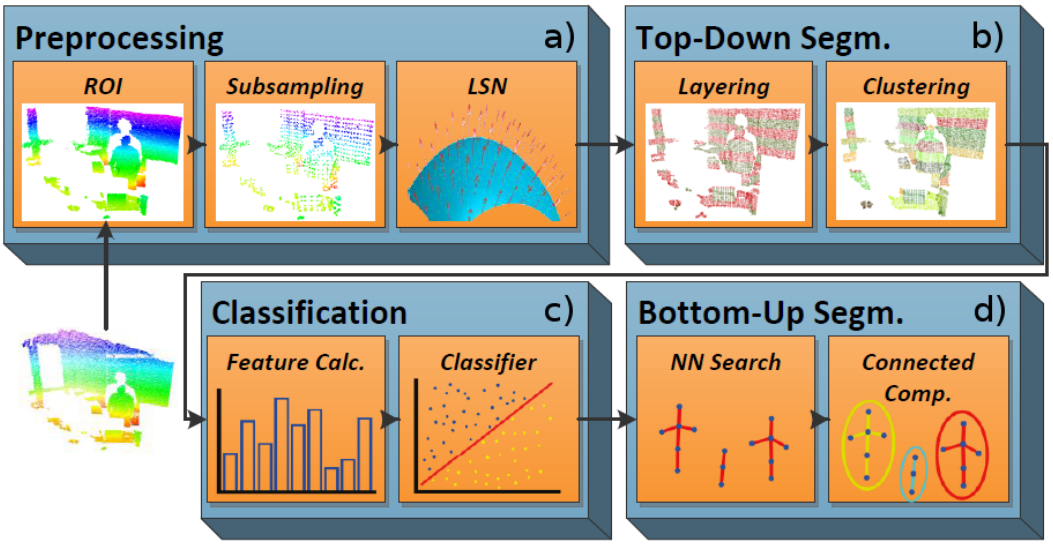


Abbildung 2.12: Ablauf des HLSN-Verfahrens nach [Hegger et al., 2013]

Datenmenge durchgeführt. Es wird eine sehr einfache Region of Interest festgelegt, die auf der maximalen Körpergröße einer Person von zwei Metern und einer gewählten maximalen Detektionsentfernung von fünf Metern besteht. Dieser Teil der Punktwolke wird nun mittels eines Voxel-Grids gesubsampled. Daraus resultierend sind von ursprünglich $\sim$ 300.000 Punkten noch $\sim$16.000 vorhanden. Für all diese Punkte werden nun die lokalen Oberflächennormalen berechnet.

Für eine möglichst schnelle **Segmentierung** (siehe Abb. 2.12 b))der Punktwolke in einzelne Cluster verwenden [Hegger et al., 2013] einen Top-Down-Ansatz. Die Punktwolke wird der Höhe nach in acht Ebenen à 25cm eingeteilt, welche dann jeweils über die euklidische Distanz in kleinere Cluster aufgeteilt werden. Durch das vorherige Aufteilen in Ebenen sowie das euklidische Clustering werden Probleme mit teilweise verdeckten Personen vermieden, da zum Beispiel bei einer von einem Tisch verdeckten Person für beide Objekte einzelne Cluster erstellt werden und nicht ein einziges großes Cluster.

Der Featurevektor jedes Clusters setzt sich aus einem *Histogram of Local Surface Normals (HLSN)* sowie der Tiefe und Breite des Clusters zusammen (siehe Abb. 2.12 c)). Für das HLSN werden drei Histogramme bestehend aus elf Bins gebildet.

Poses	Detection Rate
standing	87,29%
sitting	74,94%
part. occl.	82,35%

Motions	Detection Rate
not moving	87,29%
rnd. walking	86,32%
rnd. run	86,71%

Tabelle 2.2: Detektionsraten bei unterschiedlichen Posen und Bewegungen
*Klassifikation nach [*Hegger *et al., 2013]*

Jedes Histogramm beinhaltet eine Achse der Normalenvektoren jedes Punktes. Nach Bildung der drei Histogramme werden diese aneinandergehängt. Die letztendliche **Klassifikation** eines Clusters wird nun durch einen trainierten Random Forest durchgeführt, der in den Tests von [Hegger et al., 2013] im Vergleich mit einer SVM und AdaBoost die besten Ergebnisse erzielte.
In der abschließenden Phase werden mittels einer Bottom-Up-Segmentierung (siehe Abb. 2.12 d)) die Klassifikationsergebnisse der einzelnen Cluster zu einer Gesamtdetektion zusammengeführt. Dabei werden alle Zentren derjenigen Cluster verbunden, die einen maximalen Abstand von $2 \cdot$ *Layerhöhe* besitzen. Ein Objekt wird genau dann als Person detektiert, wenn mindestens drei Cluster des Objektes als Teil einer Person klassifiziert wurden.

Ein typischer Oberkörper: Tiefentemplate basierte Personendetektion

Die Grundidee des von [Jafari et al., 2014] vorgestellten Verfahrens besteht in der Anwendung eines Templates eines typischen Oberkörpers. Dieses Template wird z.B. mittels des Sliding-Window-Verfahrens über das Bild geschoben. An jeder Stelle wird die Differenz zwischen dem Template und dem aktuellen Bildausschnitt berechnet. Bei einem geringen Unterschied wird dementsprechend ein Bereich als Oberkörper einer Person klassifiziert. Dieses wird bis zu einer Entfernung von 5 Metern angewandt, bei größeren Entfernungen wird ein *groundHOG-Detektor* als farbbildbasierter Detektor (siehe [Sudowe und Leibe, 2011]) angewandt. [Jafari et al., 2014] Detektionssystem erreicht eine Geschwindigkeit von bis zu 43 Bildern pro Sekunde (ohne die Anwendung

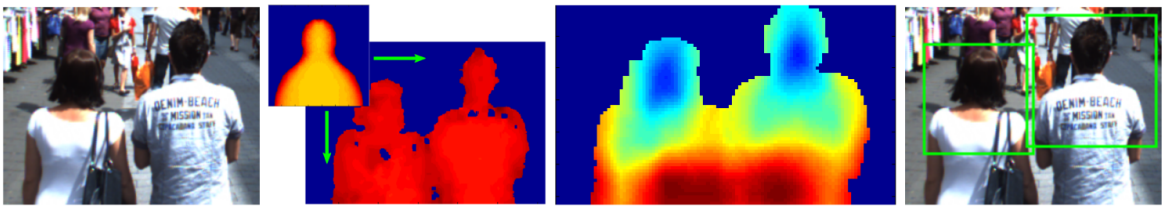


Abbildung 2.13: Oberkörperdetektion mittels eines Tiefentemplates nach [Jafari et al., 2014]
(a): Zurückprojizierte 3D-ROI, (b): Tiefentemplate, (c): korrespondierende Distanzmatrix, (d): Oberkörperdetektionen

des *groundHOG-Detektor*). Dieses wäre mit einem Sliding-Window-Verfahren nicht möglich, da das Template an sehr vielen Stellen in mehreren Skalierungen angesetzt werden müsste. Stattdessen werden aus den 3D-Tiefendaten ROIs abgeleitet, wodurch das gelernte Oberkörpertiefentemplate lediglich an wenigen Stellen im Tiefenbild angesetzt werden muss.

Generalized Christmas Trees als Repräsentation von Objekten

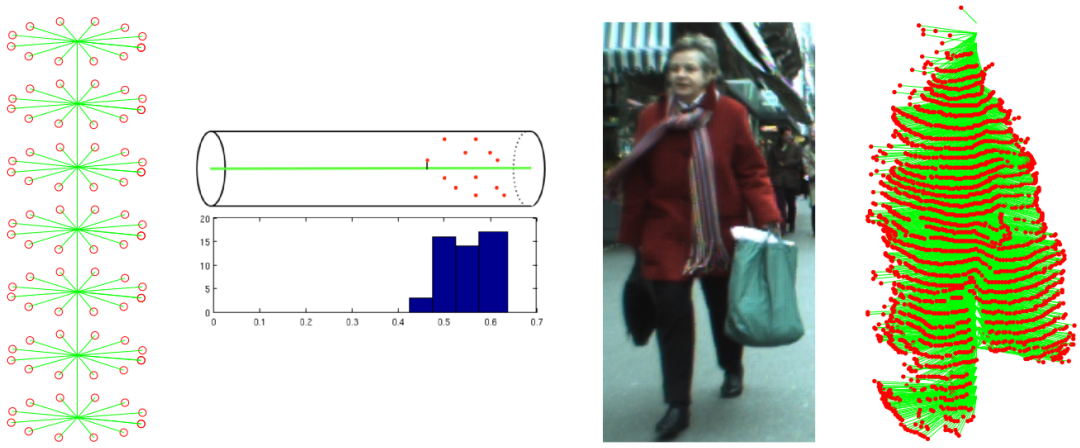
[Mitzel und Leibe, 2012] verfolgen einen *tracking-before-detection*-Ansatz. Im Unterschied zu *tracking-by-detection* wird nicht zuerst eine Detektion von z.B. Personen durchgeführt und dann einem Tracker übergeben, sondern es werden segmentierte Objekte getrackt und im Anschluss klassifiziert.
In einem ersten Schritt werden ROIs segmentiert, in denen im Anschluss GCT, gebildet werden. Hierzu wird zunächst eine Schätzung der Grundebene vorgenommen, auf die nun alle Punkte bis zu einer Höhe von zwei Metern projiziert werden. Diese werden nun in einem 2D-Histogramm gesammelt und nach ihrer quadratischen Entfernung zum Kameraursprung gewichtet, um die variierende Tiefenauflösung zu kompensieren. [Mitzel und Leibe, 2012] betrachten dabei keine Regionen, die durchgehend eine größere Höhe als zwei Meter aufweisen. Nach einer Rauschreduzierung durch einen Schwellwert werden die verbliebenen Histogrammbins in Zusammenhangskomponenten nach ihrer 8-Nachbarschaft eingeteilt. Da die Projektion von nah zusammengehenden Personen sehr eng zusammenliegt, sind die Komponenten in der Projektion miteinan-

der verbunden. Um diese weiter zu trennen, wird der Quick-Shift-Algorithmus nach [Vedaldi und Soatto, 2008] angewandt und es ergeben sich die Zusammenhangskomponenten der entsprechenden Personen. Diese werden nun auf das 2D-Kamerabild projiziert und mittels des DPM-Detektors von [Felzenszwalb et al., 2010] als Person oder Nicht-Person klassifiziert. Der Teilschritt der Extraktion von ROIs entspricht dabei dem Verfahren von [Jafari et al., 2014], welcher im eigenen Ansatz in Abschnitt 4.2.3 beschrieben wird.

Nicht nur Personen zu erkennen, sondern auch weitere Details zu erfahren, sind im Rahmen der mobilen Robotik von Bedeutung. Um zu erkennen, ob eine Person zum Beispiel große Einkaufstaschen trägt oder einen Kinderwagen schiebt, repräsentieren [Mitzel und Leibe, 2012] die Form eines Objektes über ein *Generalized Christmas Tree (GCT)*-Modell. Ein GCT besteht aus einer vertikalen Achse mit vielen Schichten, wobei jede Schicht aus gleichmäßig verteilten von der Achse ausgehenden Strahlen besteht (siehe Abb. 2.14(a)). Pro Frame werden für jeden Strahl alle Punkte in einer zylindrischen Umgebung betrachtet und derjenige Punkt ausgewählt, der den geringsten Abstand zum Strahl besitzt. Der Abstand zwischen der vertikalen Achse und der Projektion des ausgewählten Punktes auf den Strahl wird in einem Histogramm gespeichert (siehe Abb. 2.14(b)). Ein Vergleich der gelernten Modelle (siehe Abb. 2.15(a)) mit getrackten Modellen aus der Anwendungsphase liefert eine Schätzung, welche Teile des Clusters nicht dem Modell einer Person entsprechen (siehe Abb. 2.15(b) und 2.15(c)).

Erkennung von Personen in NDT-Clustern

Am Fachgebiet Neuroinformatik und Kognitive Robotik wurde ein Detektionssystem für gestürzte Personen entwickelt. Der Ablauf der Erkennung ist in Abb. 2.16 dargestellt. Zunächst wird in dem Tiefenbild die Grundebene bestimmt und entfernt, da diese keine relevanten Informationen enthält und das in einem späteren Schritt durchgeführte Clustering vereinfacht. Im Anschluss wird aus dem Tiefenbild ein 3D-Modell erstellt. Dabei wird in [Lewandowski et al., 2017] die NDT-Karte ([Magnusson et al., 2007]) als kompakte Datenrepräsentation gewählt.



(a) (b) (c) (d)

Abbildung 2.14: Generalized Christmas Tree (GCT) als Objektrepräsentation
*(a): Modell eines GCT, (b): über die Zeit akkumuliertes Histogramm, (c): 2D-Bild einer Person des Testdatensatzes, (d): GCT der Person aus (c) (nicht alle Schichten sind visualisiert), Bilder: [*Mitzel *und* Leibe*, 2012]*

(a) (b) (c)

Abbildung 2.15: Detektion von getragenen Objekten
*(a): gelerntes Personenmodell als GCT, (b): Ergebnis des Vergleich zwischen gelerntem Personenmodell und getracktem Modell in der Anwendungsphase, (c): rot markiert sind detektierte Gegenstände, die von einer Person getragen werden, Bilder: [*Mitzel *und* Leibe*, 2012]*

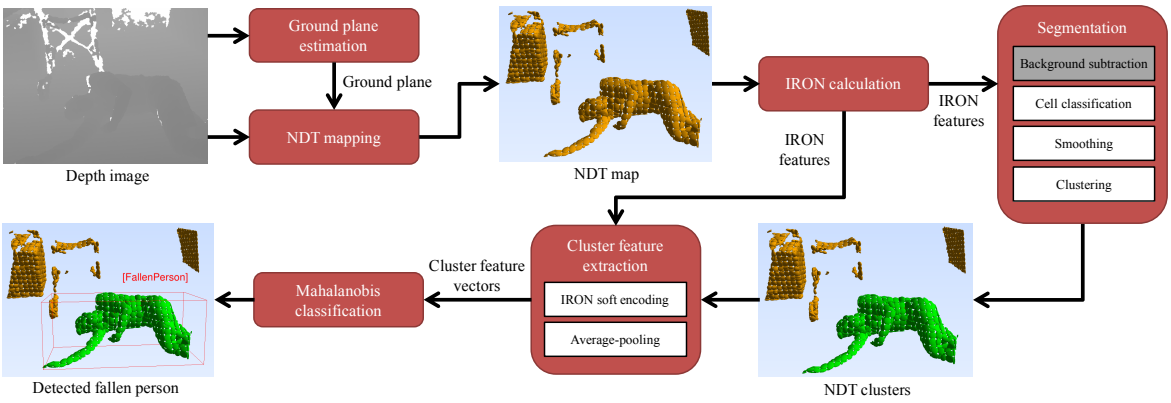


Abbildung 2.16: Übersicht über die Erkennung von gestürzten Personen nach [Lewandowski et al., 2017]

Für jede Zelle der NDT-Karte wird nun der IRON-Deskriptor (siehe 3.1.4) nach [Schmiedel et al., 2015] berechnet und im Folgenden verwendet, um eine Segmentierung durchzuführen.

Jede Zelle wird durch einen AdaBoost-Klassifikator bestehend aus mehreren Entscheidungsbäumen als Weak Learner als „von einer Person stammend“ bzw. „nicht von einer Person stammend“ eingeteilt. In einem weiteren Schritt wird das Ergebnis geglättet, um eine bessere Segmentierung zu erreichen. Zum einen werden Zellen, die fälschlicherweise als Person eingeteilt wurden, hier wieder entfernt. Zum anderen werden Cluster mit zugehörigen NDT-Zellen vervollständigt, nähere Details zu diesem Schritt sind in [Lewandowski et al., 2017] gegeben. Mittels des DBSCAN-Algorithmus ([Ester et al., 1996]) werden einzelne Cluster bestimmt, die alle nah beieinanderliegenden Zellen enthalten, die als „von einer Person stammend“ klassifiziert wurden. Diese Cluster werden im Folgenden als einzelne Objekte behandelt.

Für jede Zelle eines Clusters werden erneut die IRON-Features berechnet. Durch die Neuberechnung gehen keine Zellen mehr mit ein, die außerhalb des Clusters liegen. Nach einem *Softencoding* basierend auf einem weichen Schwellwert ([Coates und Ng, 2011]) werden alle resultierenden IRON-Features eines Clusters durch eine Mittelwertbildung in einem einzelnen Histogramm gesammelt. Über die Mahalanobisdistanz wird das Mittelwerthistogramm bewertet und die Klassenentscheidung getroffen.

2.2.4 Personendetektion mittels Deep Learning

Deep Learning-Ansätze haben im Laufe der letzten Jahre viele sehr gute Ergebnisse im Bereich der Klassifikation von Bildern erzielt. Auch für die Verwendung von Tiefendaten existieren Ansätze, die im Folgenden vorgestellt werden.

VoxNet: Ein 3D CNN zur Objekterkennung

[Maturana und Scherer, 2015] stellten 2015 das *VoxNet* als Möglichkeit vor, Deep Learning auf Punktwolken zur Objekterkennung zu verwenden. Als Input dient dabei ein Segment einer Punktwolke, das aus Segmentierungsansätzen wie in Abschnitt 2.2.3 vorgestellt oder aus einem „sliding box“Ansatz stammen kann. Ein Segment wird nun in die Datenstruktur eines *Occupancy Grids* überführt. Hierfür untersuchten [Maturana und Scherer, 2015] drei unterschiedliche Ansätze.

- **Binary Occupancy Grid**
  Bei diesem Modell besitzt jeder Voxel einen binären Status *frei* oder *belegt*.

- **Density Grid**
  Jeder Voxel besitzt eine kontinuierliche Dichte, die korrespondierend zur Wahrscheinlichkeit ist, dass dieser Voxel einen Sensorstrahl blockt (*Belegtheitswahrscheinlichkeit*).

- **Hit Grid**
  Dieses Modell ignoriert den Unterschied zwischen unbekanntem und leerem Raum. Es wird pro Voxel lediglich gezählt, wie oft jeder Voxel von einem Sensorstrahl getroffen wird. In dem zugehörigen Paper wurden trotz des hierdurch entstehenden Informationsverlustes gute Ergebnisse erzielt.

Die berechneten Occupancy Grids dienen nun als Input für ein *3D Convolutional Neural Network (CNN)* bestehend aus fünf Schichten. Die Struktur des verwendeten CNN ist in Abb. 2.17 dargestellt.
Ein Nachteil des Verfahrens ist eine deutlich erhöhte Klassifikationsdauer bei einer hohen Anzahl an Punkten (z.B. bei Verwendung der Kinect2), da für die Berechnung

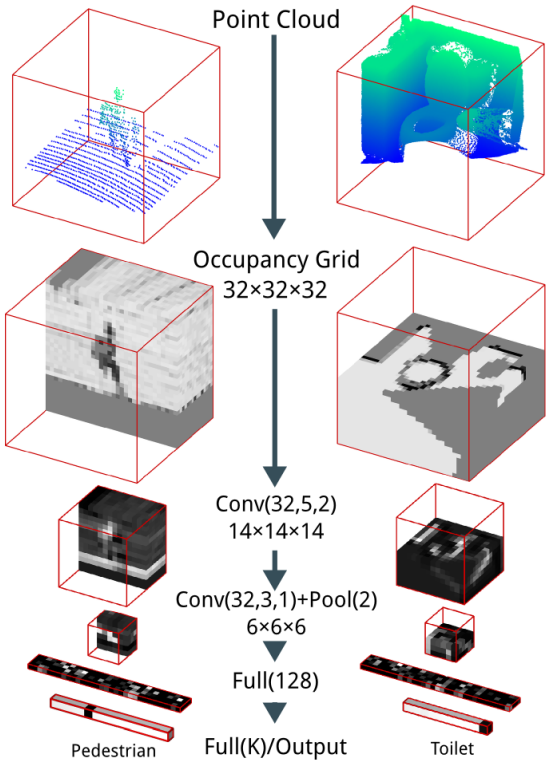


Abbildung 2.17: Architektur des VoxNets nach [Maturana und Scherer, 2015] *Conv(f,d,s): Convolutional Layer mit vierdimensionalem Input (drei räumliche Dimensionen und die Feature Maps); Pool(m): Pooling Layer Downsampling des Input um Faktor m; Full(n): Fully Connected Layer mit n Outputneuronen*

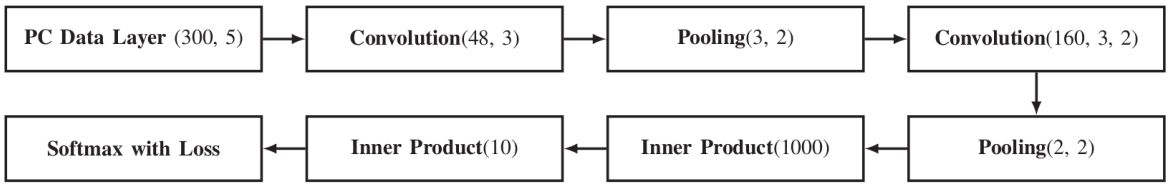


Abbildung 2.18: Architektur des PointNets nach [Garcia-Garcia et al., 2016]

der *Occupancy Grids* ein Raytracing erforderlich ist. Verbessert werden kann dies z.B. durch die Verwendung des *Hit Grid Modells*.

PointNet

Inspiriert durch das zuvor beschriebene *VoxNet* (siehe Abschnitt 2.2.4) entwarfen [Garcia-Garcia et al., 2016] das *PointNet*. Hiermit sollen die guten Klassifikationsergebnisse des *VoxNet* erhalten bleiben, jedoch eine bessere Klassifikationsgeschwindigkeit erreicht werden.

In einem Vorverarbeitungsschritt wird die Punktwolke auf die Größe des Grids skaliert. Das *Occupancy Grid* fester Größe wird dann berechnet durch eine Projektion jedes Punktes der Punktwolke in das entsprechende Voxel. Jedes Voxel enthält nun die Anzahl der enthaltenen Punkte und wird nach allen Projektionen normalisiert. Das *PointNet* besteht aus mehreren Schichten, die in Abb. 2.18 dargestellt sind. Im Point Cloud Data Layer wird die Input-Punktwolke in die Occupancy-Grid-Struktur umgewandelt, dabei entspricht der erste Parameter der Gridgröße in Breite, Höhe und Tiefe. Der zweite Parameter definiert die Größe eines einzelnen Voxels. Die Convolutional Layer sind analog zu denen des 2.2.4 definiert. Die Pooling Layer enthalten als Parameter die Filtergröße und die Schrittweite. Inner Product Layer entsprechen den vollverschalteten Layern mit einer festen Anzahl an Neuronen.

2.3 Zusammenfassung

Es existieren zahlreiche Verfahren zur Detektion von Personen in Tiefendaten. Diese basieren am häufigsten auf der Verwendung von Tiefenbildern, für Punktwolken gibt es

weniger Entwicklungen. Der „Sliding-Window-Ansatz“ wird wie in 2D-Bilddaten ebenso in Tiefenbildern genutzt, in Punktwolken ist ein ähnlicher Ansatz als „sliding-Voxel“ insbesondere im vorliegenden Einsatzszenario nicht möglich.
Die dargestellten Verfahren zur Detektion von Personen in Punktwolken (siehe Abschnitt 2.2.4) zeigen, dass ein Ansatz basierend auf Deep Learning möglich ist. Hierfür ist jedoch eine sehr große Anzahl an Trainings- und Testdaten notwendig. Da zu Beginn dieser Masterarbeit keine Daten aus dem Anwendungsszenario vorhanden sind und die Erstellung der benötigten Datensätze einen großen Teil dieser Arbeit darstellt, wird kein Ansatz basierend auf Deep Learning verfolgt. Dies kann mit dem in dieser Masterarbeit entstehenden Datensatz in zukünftigen Arbeiten untersucht werden.
Der schichtenbasierte Ansatz von [Martinson und Yalla, 2016] und die Segmentierung von Clustern aus einer Punktwolke von [Jafari et al., 2014] bieten eine gute Ausgangsposition für das zu entwickelnde System zur Detektion von stehenden und hockenden Personen in einer Einkaufsmarktumgebung.

Kapitel 3

Theoretische Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen, die im eigenen Ansatz verwendet werden, dargestellt. Zunächst werden die verwendeten Features und ihre Eigenschaften erläutert. Im Rahmen dieser Masterarbeit werden Features für Punktwolkencluster berechnet zur anschließenden Klassifikation. Des Weiteren werden die ausgewählten Detektionssysteme vorgestellt, welche in der Evaluierung des eigenen Ansatzes als Referenzverfahren dienen.

3.1 Features in 3D-Punktwolken

3.1.1 SHOT: Signature of Histograms of Orientations

Der „Signature of Histograms of Orientations“-Deskriptor wurde von [Tombari et al., 2010] als sehr aussagekräftig und schnell berechenbar vorgestellt.
Zu jedem Anfragepunkt besteht die lokale Nachbarschaft aus einer Sphäre, die an ein lokales Koordinatensystem orientiert ist. Diese wird nun in einzelne Zellen aufgeteilt (siehe Abb. 3.1) und pro Zelle ein Histogramm gebildet, welches den Kosinus des Winkels zwischen dem Normalenvektor des Anfragepunktes sowie der Normale jeden Punktes in Bins einteilt. Um mögliche unterschiedliche Punktdichten auszugleichen, werden die Histogramme auf 1 normalisiert. Die Aufteilung der Sphäre findet dabei in acht Abschnitte in Azimuthrichtung, zwei Abschnitte in der Höhe und zwei radiale

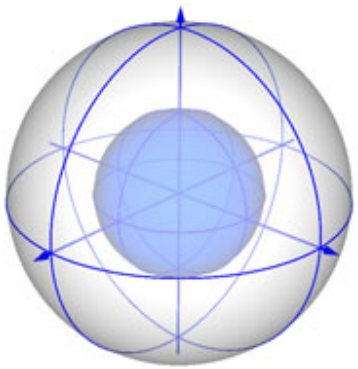


Abbildung 3.1: Aufteilung der Sphäre zur Bildung des SHOT-Deskriptors nach [Tombari et al., 2010]

Aufteilungen statt. Durch die Aufteilung ergeben sich 32 Zellen. Bei einer Binanzahl von 11 pro Histogramm nach [Tombari et al., 2010] ergibt sich somit eine Länge des resultierenden Featurevektors von 352.
In [Tombari et al., 2011] wird der SHOT-Deskriptor erweitert, indem die Farbe der einzelnen Punkte berücksichtigt wird. Hierfür werden zunächst die RGB-Farbwerte in den CIELAB-Farbraum konvertiert. Pro Zelle wird ein Histogramm mit 31 Bins gebildet mit der L1-Distanz zwischen dem Anfragepunkt und allen in der Zelle enthaltenen Punkte. Hieraus ergibt sich eine Featurevektorlänge für den sogenannten Color-SHOT von 1344. Dieser Ansatz ist im Rahmen dieser Masterarbeit insbesondere interessant für den optionalen Einbezug von Farbe.

3.1.2 FPFH: Fast Point Feature Histogram

Das Fast Point Feature Histogram (FPFH) (entworfen von [Rusu et al., 2009]) basiert auf dem Point Feature Histogramm von [Rusu et al., 2008] und wird häufig in der Objekterkennung zum Matching von Punktwolken verwendet und benötigt Zeit $O(k)$. Im ersten Schritt wird für jeden Punkt p_q der Punktwolke die Differenz zwischen der Orientierung seines Normalenvektors und den Normalenvektoren aller k Nachbarn bestimmt. Als Nachbarn gelten hierbei alle Punkte in einem Radius r. Die Differenz wird durch ein Tripel $< \alpha, \Phi, \theta >$ beschrieben, welche das sogenannte Simplified Feature

Histogram bilden (siehe Formel 3.1).
Im zweiten Schritt wird aus allen betrachteten Punkten das Fast Point Feature Histogram aus den Simplified Point Feature Histogrammen berechnet. Der Wert des FPFH des Punktes p_q berechnet sich aus dem Wert des SPFH des Punktes plus aller Werte des SPFH der benachbarten Punkte, welche distanzbasiert mit einem Gewicht ω_k einfließen.

$$\alpha = v \cdot n_j \tag{3.1a}$$

$$\phi = (u \cdot (p_j - p_i))/\|p_j - p_i\| \tag{3.1b}$$

$$\theta = \arctan(w \cdot n_j, u \cdot n_j) \tag{3.1c}$$

$$FPFH(p) = SPF(p) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{\omega_k} \cdot SPF(p_k) \tag{3.1d}$$

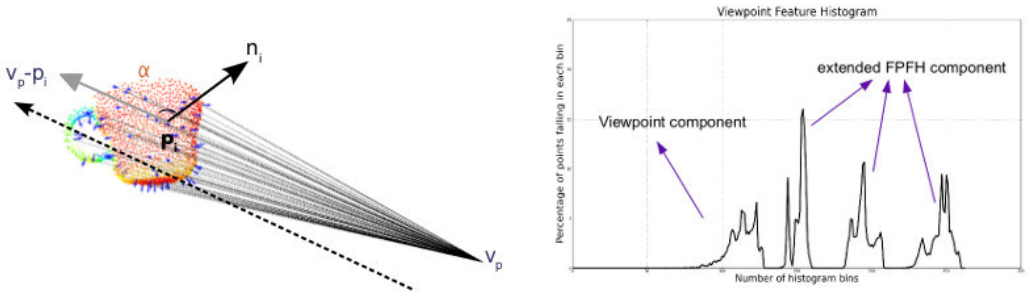
3.1.3 VFH: Viewpoint Feature Histogram

Das Viewpoint Feature Histogram (VFH) von [Rusu et al., 2010] basiert auf dem im vorigen Abschnitt beschriebenen FPFH durch Hinzufügen eines Standpunktes. Der Standpunkt kann dabei zum Beispiel dem Kamerazentrum entsprechen und somit die Blickrichtung entlang der Kamerastrahlen. Das gesamte Cluster wird aus diesem Standpunkt heraus betrachtet und zwei Komponenten für das VFH bestimmt, welche hintereinander gehängt das Viewpoint Feature Histogram ergeben:

- Ein Histogramm der Winkel zwischen der Blickrichtung aus dem Standpunkt heraus und der Normalen jedes Punktes (siehe Abb. 3.2(a))
- Ein Histogramm der Dreh-, Roll- und Gierwinkel zwischen der Standpunktrichtung im Zentrum des Clusters und jeder Normalen auf der Oberfläche (siehe Abb. 3.2(b))

Eine beispielhafte Darstellung eines Viewpoint Feature Histograms ist in Abb. 3.2 ersichtlich. Die in dieser Masterarbeit verwendete Implementierung der Point Cloud Library [1] verwendet zusätzlich ein Distanzhistogramm mit 45 Bins, in welches die

[1] `http://docs.pointclouds.org/trunk/classpcl_1_1_v_f_h_estimation.html`



(a) (b)

Abbildung 3.2: Darstellung eines beispielhaften Viewpoint Feature Histogram *(a) Berechnung des Winkels zwischen dem Standpunkt und der Normalen von p_i, v_p: Viewpoint, p_i: aktuell betrachteter Punkt der Punktwolke, $v_p - p_i$: Vektor Richtung des Viewpointvektors in p_i verschoben, n_i: Normale des Punktes p_i, α: berechneter Winkel für erstes Histogramm; (b) Darstellung der zwei berechneten Histogramme, x-Achse: Bins der Histogramme, y-Achse: prozentualer Anteil der der Punkte pro Bin; Bilder: [*Rusu *und* Cousins*, 2011]*

Distanzen zwischen dem Viewpoint und allen Punkten des Clusters gesammelt werden.

3.1.4 IRON: A Fast Interest Point Descriptor for Robust NDT-Map Matching

Das aus dem State of the Art bekannte Verfahren zur Erkennung von gestürzten Personen von [Lewandowski et al., 2017] verwendet die IRON-Features in NDT-Karten von [Schmiedel et al., 2015]. Diese Features eigneten sich nicht nur zum Map-Matching bei der Roboter-Lokalisation, sondern auch zur Detektion von gestürzten Personen. Im Rahmen dieser Masterarbeit werden aus zwei Gründen keine NDT-Karten verwendet. Zum einen handelt es sich bei der aktuell im Fachgebiet für Neuroinformatik und Kognitive Robotik verwendete Implementierung um eine Occupancy-NDT-Variante. Diese NDT-Karten werden durch über die Zeit integrierte Messungen gebildet, wodurch Freibeweise notwendig sind, wenn sich Personen bewegen. Bewegliche Objekte ziehen somit einen Schleier hinter sich her, da die NDT-Zellen hinter ihnen

nicht direkt als frei bewiesen werden können. Zusätzlich ist die vorliegende Occupancy-NDT-Implementierung nur bis zu einer Distanz von vier Metern nutzbar aufgrund der geringeren Punktdichte der Punktwolke in größeren Entfernungen. Da im vorliegenden Anwendungsszenario Personen bis 10 Metern detektiert werden sollen, kann diese Implementierung der NDT-Karten nicht verwendet werden. Die IRON-Features werden stattdessen auf der Punktwolke für jeden Punkt eines Kandidatenclusters berechnet. Der einzige Unterschied in der Berechnung der Features besteht in der Bestimmung der Nachbarschaften der Punkte. Bei NDT-Karten entspricht die Nachbarn den Punkten aus demselben Voxel Grid, bei Punktwolken handelt es sich um eine kugelförmige Nachbarschaft oder ein kNN (nähere Erläuterung des Verfahrens in [Dasarathy, 1991]). Eine detailliertere Beschreibung der Berechnung ist im Pseudocode 3.3 ersichtlich.

Eingaben

```
1   PC = {(x_1, y_1, z_1), ..., (x_n, y_n, z_n)}        // Punktwolke mit n Punkten
2   float searchRadius                                  // Suchradius für benachbarte Punkte
3   int minNeighbors                                    // min. Anz. an Nachbarn pro Punkt
4   int angleBins          // Anz. Bins für Histogramm über Winkel zwischen zwei Vektoren
5   int distanceBins       // Anz. Bins für Histogramm über Distanz zwischen zwei Punkten
6   float entropyThreshold                              // Schwellwert für minimale Entropie
```

Algorithmus

Für jeden Punkt der Punktwolke:: //

```
7       Initialisiere Histogramm mit Nullen;
8       Suche alle Nachbarn im Radius searchRadius;
9       Wenn #Nachbarn == 0, dann;
10         goto nächste Iteration
11      sonst;
12         berechne Normalenwinkel und Verbindungswinkel;
13         berechne Distanz zwischen Punkten;
14         erhöhe Histogrammeinträge;
```

Matrixnormalisierung: // **jede Matrix unabhängig voneinander**

```
15      mNrMatrixEntries = angleBins · distanceBins;
16      nrFilledDistBins = Anzahl der gefüllten Bins;
17      Für jeden Distanzbin;
18         Wenn distanceBinCount[distanceBin] > 0;
19            normiere Histogramme der Normalenwinkel und Verbindungswinkel;
```

Entropieberechnung: //

```
20      Für jeden Punkt;
21         Wenn #Nachbarn == 0, dann;
22            setze Entropie auf Null;
23         sonst;
24            entropie = berechneEntropie(Normalenwinkelhistogramm);
25         Wenn entropie < entropyThreshold;
26            entropie = 0;
```

Rückgabe

```
27  Vektor mit IRON-Deskriptoren                        // pro Punkt ein IRON-Deskriptor
```

Abbildung 3.3: Implementierung der IRON-Features in Punktwolken

3.2 Klassifikation

3.2.1 AdaBoost

Sehr berechnungseffizient bei der Klassifikation von Merkmalsvektoren sind sogenannte schwache Klassifikatoren (engl. „Weak Learner “), welche sehr schnell ausgewertet werden können, da sie häufig nur ein Merkmal berücksichtigen und anhand dessen eine Separierung der Klassen durchführen. Diese liefern jedoch bei der Klassifikation meist schlechtere Ergebnisse als umfangreichere Klassifikatoren, die mehrere Merkmale berücksichtigen. Boosting bezeichnet die Kombination von mehreren schwachen Klassifikatoren zur Bewertung eines Merkmalvektors, welches in Abb. 3.4 schematisch dargestellt wird. Hierdurch soll sowohl eine gute als auch in der Berechnung effiziente Klassifikation erreicht werden. Das bekannteste Boosting-Verfahren „AdaBoost “ wurde von Yoav Freund und Robert Schapire entworfen ([Freund und Schapire, 1997]). In jeder Iteration wird beim Training ein weiterer *Weak Learner* zu dem Ensemble hinzugefügt, welches den *Strong Learner* ergibt. Nach jedem Hinzufügen werden die Trainingsdaten, die bisher noch falsch klassifiziert werden, stärker gewichtet, sodass die darauffolgenden *Weak Learner* speziell auf diese Datenpunkte ein Augenmerk bezüglich der Klassifikation besitzen. Durch diese Änderung der Gewichte der einzelnen Beispiele handelt es sich um ein adaptives Lernverfahren.

Weak Learner

Für die Auswahl der geeigneten *Weak Learner* gibt es verschiedene Möglichkeiten. So wurden z. B. von Viola und Jones ([Viola und Jones, 2001]) einfache Weak Learner verwendet, die jeweils nur ein Merkmal f und einen Schwellwert Θ besaßen (siehe Gleichung 3.2a), um die Zugehörigkeit eines Datenpunktes zu einer Klasse zu bestimmen. Der *Strong Learner* trifft die Entscheidung der Klassenzugehörigkeit durch die Klassifikationen der *Weak Learner* und deren Gewichtung (siehe Gleichung 3.2b).
Die *Weak Learner* können jedoch auch aus anderen Klassifikatoren bestehen, zum Beispiel aus *Decision Trees*. Bei *Decision Trees* werden die Trainingsdaten in Subdatensätze anhand eines Attributtests (Schwellvertvergleich, binärer Vergleich) des Featurevektors aufgeteilt. Auf jeden Subdatensatz wird dieses Verfahren rekursiv an-

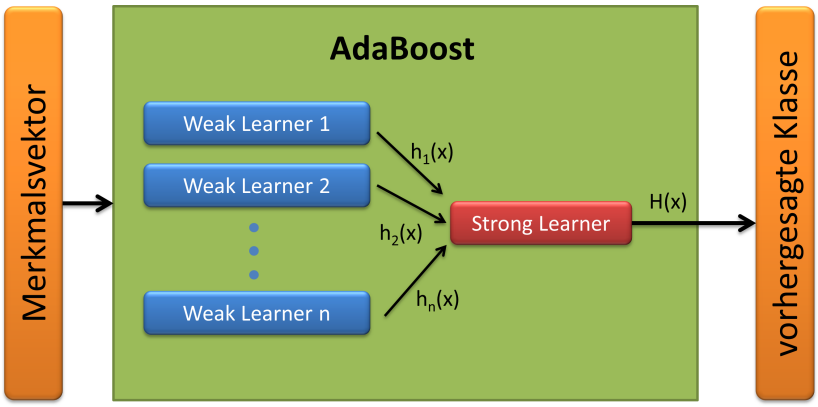


Abbildung 3.4: Schema des AdaBoost-Verfahrens

gewandt, sodass sich eine Baumstruktur ergibt. Die Kombination von AdaBoost mit *Decision Trees* wurde führte in [Weinrich et al., 2014] zu guten Ergebnissen bei der Klassifikation von Personen in 2D-Laserscans. Im Rahmen dieser Masterarbeit werden daher auch *Decision Trees* als Weak Learner verwendet.

$$h_t(x) = \begin{cases} +1 & \text{falls } f_t(x) < \Theta_t \\ -1 & \text{sonst} \end{cases} \tag{3.2a}$$

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t * h_t(x)\right) \tag{3.2b}$$

3.2.2 Support Vector Machine (SVM)

Ein viel verbreiteter Klassifikator wurde in den frühen 90er Jahren zunächst von [Boser et al., 1992] vorgestellt. Das Ziel einer SVM ist die Berechnung einer Hyperebene, welche die Featurevektoren der Trainingsdaten so trennt, dass der Abstand der der Hyperebene am nächsten gelegenen Inputvektoren maximiert wird. Es gibt zwei verschiedene Arten von SVMs: lineare und nichtlineare SVMs. Linear separierbare Daten können durch eine einfache *gerade* Hyperebene getrennt werden. Solch eine lineare Trennung ist jedoch nicht in allen Anwendungsszenarien gegeben. Um nicht linear separierbare Daten mittels eine Hyperebene zu trennen, wird der gesamte Vektorraum (und damit alle Vektoren) in einen höherdimensionaleren Raum überführt. In einem

Raum mit bis zu unendlich vielen Dimensionen ist jede Vektormenge linear trennbar (siehe Abb. 3.5(b)). Durch eine Rücktransformation wird die Hyperebene im niedrigdimensionalen Raum zu einer nicht linearen Trenngerade (siehe. Abb. 3.5(d)). Diese Transformationen zu berechnen, ist jedoch mit einem deutlich erhöhten Berechnungsaufwand verbunden. Um dieses zu umgehen, wird bei SVMs der sogenannte *Kernel-Trick* verwendet. Hierbei wird die Transformation jedes Vektors in den hochdimensionalen Raum nicht berechnet, denn sowohl beim Training als der Anwendung einer SVM werden die transformierten Vektoren nicht benötigt. Lediglich das Skalarprodukt der transformierten Vektoren ist von Bedeutung und muss berechnet werden. Mittels positiv definiter Kernelfunktionen wie z.B. des RBF-Kernels ist die Berechnung des Skalarproduktes zweier transformierter Vektoren im hochdimensionalen Raum möglich, ohne die Transformation zu berechnen. Somit ist eine effiziente Entscheidung möglich, auf welcher Seite der Hyperebene ein zu klassifizierender Feature-Vektor liegt.

3.2.3 2-Klassen-SVM vs. Multi-Klassen-SVM

Bei der 2-Klassen-SVM existieren genau zwei Klassen, eine für Positivdaten sowie eine für Negativdaten. In vielen Anwendungsfällen ist jedoch eine Unterscheidung zwischen mehr als zwei Klassen notwendig. So könnte ein mobiler Roboter zum Beispiel im vorliegenden Anwendungsszenario nicht nur Personen detektieren, sondern auch ihre Pose (stehende Person und hockende Person als getrennte Klassen). Hierfür existieren in der Literatur zwei Ansätze, welche sich in der Anzahl der zu trainierenden Klassifikatoren unterscheiden. Zur Unterscheidung von N Klassen wird bei der *One-vs.-all* Strategie für jede Klasse eine SVM trainiert (in Summe N SVMs). Für jede SVM wird dabei eine Klasse als die Positivklasse gewertet und alle anderen zu einer Negativklasse zusammengefasst. Im zweiten Ansatz werden $\frac{N\cdot(N-1)}{2}$ Klassifikatoren bestimmt, dabei wird pro mögliches Klassenpaar eine eigene SVM trainiert. In der Anwendungsphase wird über eine Mehrheitsentscheidung[2] das Klassifikationsergebnis festgelegt.

[2]Bei Gleichstand der Votings existieren mehrere Möglichkeiten. In der im Rahmen dieser Masterarbeit verwendeten OpenCV-Implementierung [OPENCV, 2017] wird bei Gleichstand das Klassenlabel der Klasse mit dem geringeren Index gewählt.

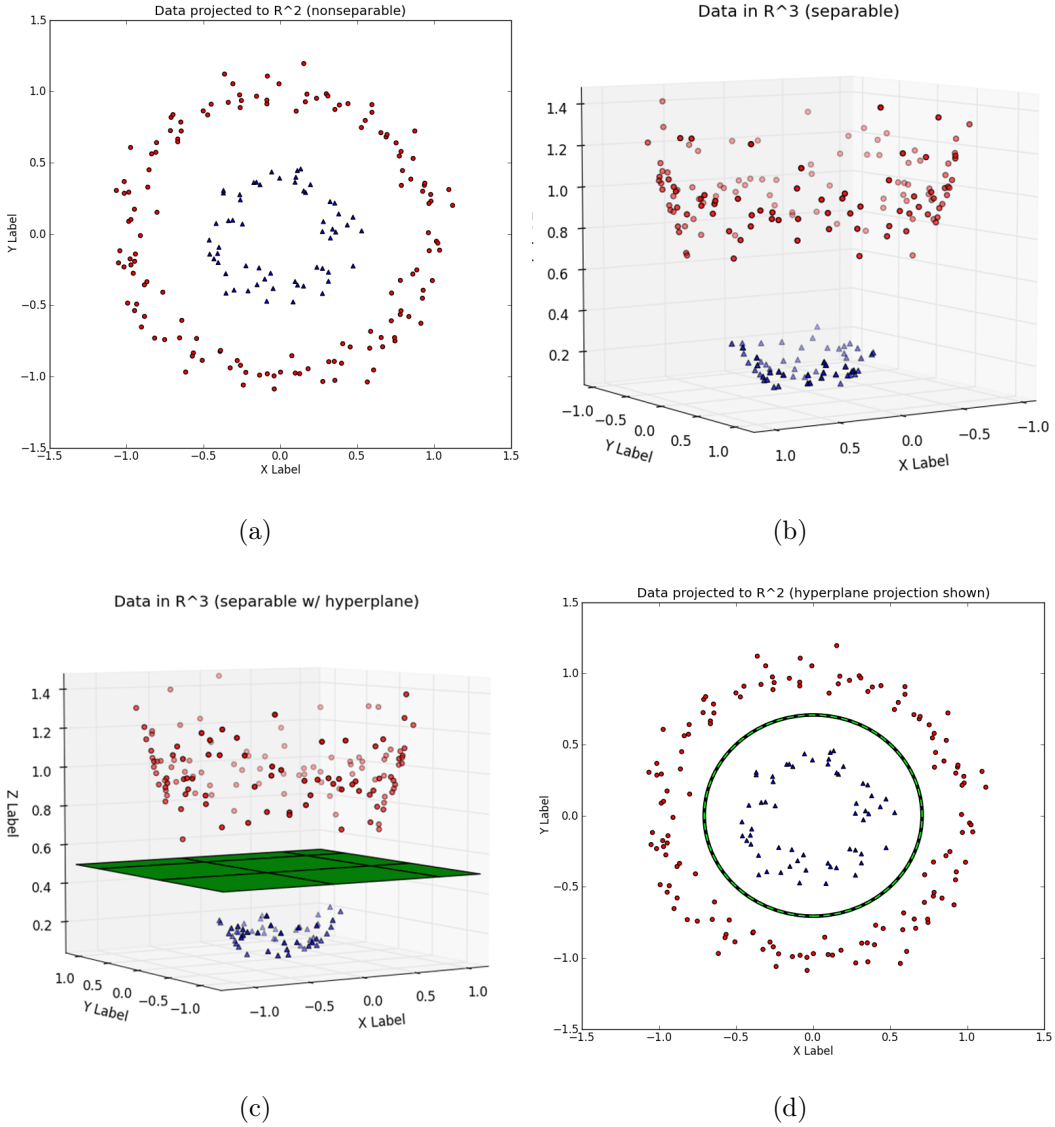
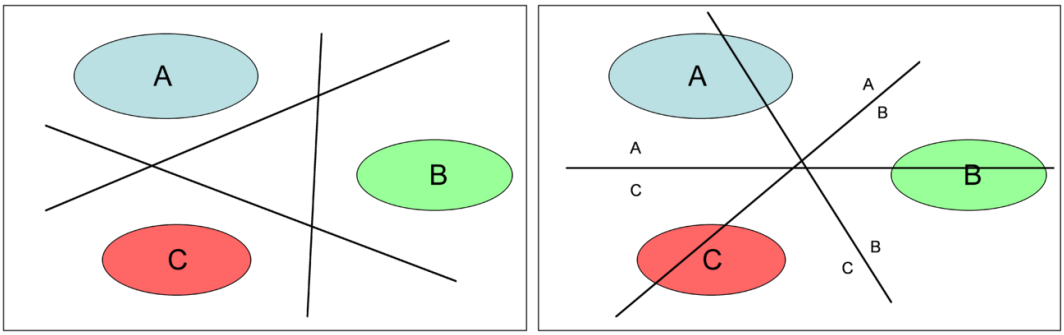


Abbildung 3.5: Transformtion in höherdimensionalen Raum zur Trennung der zwei Klassen durch eine lineare Hyperebene

(a) nicht linear separierbares Klassifikationsproblem, (b) Projektion der Inputvektoren in Featurespace, (c) Trennung der Inputvektoren durch lineare Hyperebene im Featurespace, (d) Rückprojektion der linearen Hyperebene zur Darstellung der korrekten Klassentrennung; Bilder: `http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`



(a) One-vs.-All (b) One-vs.-One

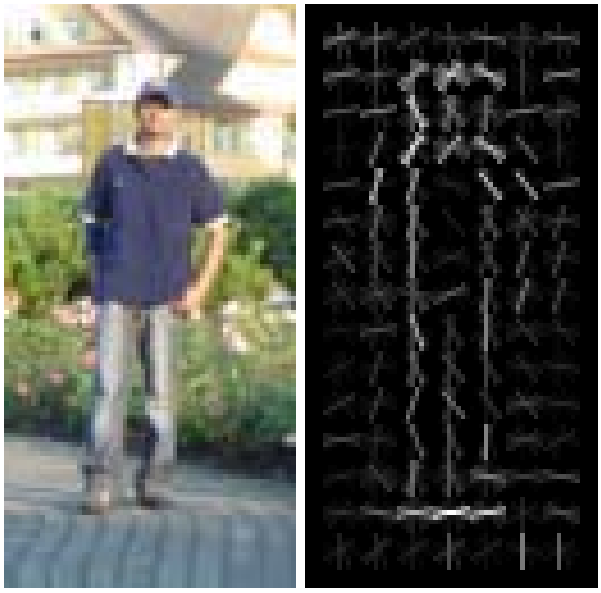
Abbildung 3.6: Vergleich Multi-Klassen-SVMs
Quelle: `http://courses.media.mit.edu/2006fall/mas622j/Projects/aisen-project/`

3.3 Detektionssysteme zum Vergleich

In der Literatur existieren zahlreiche Verfahren zur Detektion von Personen in 2D- und 3D-Bildern. Am Fachgebiet für Neuroinformatik und Kognitive Robotik existieren Implementierungen von mehreren Detektionssystemen für RGB-Bilder, die im Rahmen dieser Masterarbeit zum Vergleich herangezogen werden. Die Grundidee und Funktionsweise der Verfahren wird im Folgenden kurz dargestellt, für nähere Informationen der Verfahren für 2D-Farbbilder sei an dieser Stelle auf entsprechende Literatur verwiesen.

3.3.1 Tiefentemplate basierte Persondendetektion

Die öffentlich verfügbare Implementierung des Personendetektors basierend auf einem gelernten Tiefentemplate wurde für die Verwendung am Fachgebiet für Neuroinformatik und Kognitive Robotik angepasst und als ein Referenzverfahren verwendet. Die Funktionsweise wird in Abschnitt 2.2.3 erläutert.

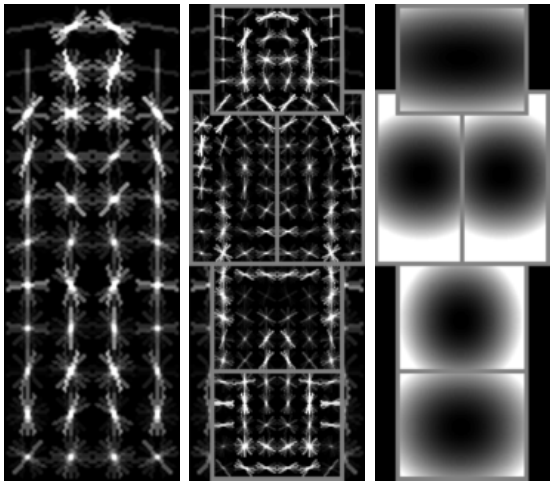


(a) (b)

Abbildung 3.7: Beispiel für einen HOG-Deskriptor anhand einer Person
*(a) Ausschnitt aus RGB-Bild, (b) Visualisierung der Elemente des Featurevektors, die für die Klasse Person entscheidend sind; Bilder: [*Dalal *und* Triggs*, 2005]*

3.3.2 HOG: Histograms of Oriented Gradients

[Dalal und Triggs, 2005] entwickelten den HOG-Deskriptor aufgrund der Grundidee, dass das lokale Aussehen und die Form von Objekten gut durch die Verteilung von lokalen Gradientenstärken und Kantenrichtungen beschrieben werden kann. Dafür wird in einem Vorverarbeitungsschritt jedes Bild normalisiert mit verschiedenen Varianten (Farbraum, Gammakorrektur etc.). Dann wird der zu klassifizierende Bildausschnitt in Zellen aufgeteilt und für jede Zelle ein Histogramm der Gradientenrichtungen berechnet. Durch eine Verkettung sämtlicher Histogramme eines Detektionsfensters ergibt sich der HOG-Deskriptor. Für die Klassifikation verwenden [Dalal und Triggs, 2005] eine Support Vector Machine (siehe 3.2.2). Bei der Betrachtung des Deskriptors (siehe Abb. 3.7(b)) ist die Grundidee ersichtlich, da sich die Grundrisse einer Person anhand der Gradienten erkennen lässt.



(a) (b) (c)

Abbildung 3.8: Komponentenmodell des Part-HOG-Detektors
*(a) grobkörniger Wurzelfilter, (b) Teilfilter für höhere Auflösungsstufen, (c) räumliches Modell zur Lage der Körperteile; Bilder: [*Felzenszwalb *et al., 2010]*

3.3.3 Part-HOG: Object Detection with Part Based Models

Der von [Felzenszwalb et al., 2010] entworfene Part-HOG-Detektor nutzt zur Klassifikation von Objekten eine Mischung aus unterschiedlichen Filtern, die auf verschiedenen Stufen der Skalierungspyramide des Bildes angewandt werden. Auf einer geringen Skalenstufe wird ein grobkörniger, sogenannter *Wurzelfilter* angewandt, der in der Personendetektion ein Modell des gesamten Körpers enthält (siehe Abb. 3.8(a)). Auf einer höheren Auflösung werden partielle Modelle verwendet, die nur einzelne Körperteile enthalten wie zum Beispiel den Kopf oder den Schulterbereich (siehe Abb. 3.8(b)). Von den Scores der Filterantworten der partiellen Modelle werden zunächst die Deformationskosten abgezogen, welche von der relativen Position der partiellen Modelle zum Wurzelfilter abhängen. Die Summe der übrig gebliebenen Scores bildet den Score der Hypothese. Die Implementierung des am Fachgebiet für Neuroinformatik und Kognitive Robotik vorhandenen Part-HOG-Detektors basiert zusätzlich auf einer Erweiterung durch [Dubout und Fleuret, 2012], welche eine beschleunigte Berechnung des Detektionsscores ermöglicht.

3.3.4 FPDW: The Fastest Pedestrian Detector in the West

[Dollar et al., 2010] verwenden in ihrem Ansatz zur Personendetektion das Prinzip der Channel Features, welches von [Dollar et al., 2009] vorgestellt wurde. Ein Channel entspricht einem registrierten Kanal des Originalbildes. So kann zum Beispiel jeder Farbkanal eines RGB-Bildes als einzelner Channel gewählt werden. Weitere Channels werden durch Transformationen des Eingangsbildes gebildet, [Dollar et al., 2010] verwenden unter anderem Gradienten ähnlich zu [Dalal und Triggs, 2005].
Eine hohe Bildrate erreichen [Dollar et al., 2010] durch den im Folgenden erläuterten Ansatz. Mehrere Verfahren wie zum Beispiel [Cho et al., 2012] und [Lowe, 2004] verwenden Skalierungspyramiden von Bildern zur Detektion von Personen unterschiedlicher Größe und Entfernung. Die Berechnung der jeweiligen Pyramiden und Features in allen Auflösungsstufen erfordert einen hohen Berechnungsaufwand, wodurch die Personendetektion meist nur in einer geringen Framerate möglich ist. [Dollar et al., 2010] verwenden daher einen hybriden Ansatz, welcher eine nur spärliche Auflösungspyramide des Eingangsbildes berechnet und zwischen diesen Auflösungsstufen eine Klassifikatorpyramide verwendet. In ähnlichen Auflösungen können die Features in den herunterskalierten Bildern effizient und ausreichend robust approximiert werden. Dieses ist deutlich effizienter als der herkömmliche Ansatz, die Features komplett neu zu berechnen. Trotz der Approximation erreichen [Dollar et al., 2010] eine nur um $1 - 2\%$ geringe Detektionsgenauigkeit als andere vergleichbare Methoden bei einer Beschleunigung der Detektionszeit um Faktor 10-100 je nach zu vergleichender Methode. Abb. 3.9 zeigt einen Vergleich der drei Varianten zur Detektion von Personen unterschiedlicher Größe oder Entfernung.

(a) (b) (c)

Abbildung 3.9: Hybrider Ansatz zur Pyramidenbildung beim FPDW
*grün: Klassifikator in Originalgröße, gelb: herunterskalierter Klassifikator, blau: hochskalierter Klassifikator; (a) klassische dichte Skalierungspyramide der Bilder (die Größe des Klassifikatorfensters bleibt gleich), (b) Skalierungsstufen für den Klassifikator (die Auflösung des Bildes bleibt gleich), (c) hybrider Ansatz mit einer spärlichen Bildpyramide mit einer Skalierungsweite von einer Oktave und innerhalb einer Oktave jeweils eine Klassifikatorpyramide; Bilder: [*Dollar *et al., 2010]*

Kapitel 4

Punktwolkenbasierter Supermarkt-Personendetektor

In diesem Kapitel wird das im Rahmen dieser Masterarbeit entworfene Detektionssystem zur Erkennung von hockenden und stehenden Personen in einer Einkaufsmarktumgebung vorgestellt. Da es sich um einen Detektor für Personen in einem Supermarkt handelt, erhält das System den Namen **Su**permarkt-**Per**sonen-Detektor, in kurz SuPer-Detektor. Abschnitt 4.1 zeigt einen kurzen Überblick über das System, welches in den folgenden Teilen näher beschrieben wird. Die Vorverarbeitung der Daten wird in Abschnitt 4.2 dargestellt. Dort wird zum einen eine Einteilung möglicher Körperhaltungen in Klassen (siehe Abs. 4.2.1) und die Segmentierung von Kandidatenclustern aus den Tiefenbildern (siehe Abs. 4.2.3) erläutert. Zum anderen werden Möglichkeiten aufgezeigt, die Performanz des Systems zu erhöhen (siehe Abs. 4.2.4) und sensorische Eigenschaften zu behandeln (siehe Abs. 4.2.5). Anschließend erfolgt eine Beschreibung der Feature-Extraktion in Abschnitt 4.3 sowie des Trainings unterschiedlicher Klassifikatoren.

4.1 Systemübersicht

Die Systemarchitektur des im Rahmen dieser Masterarbeit entworfenen Detektionssystems ist in Abbildung 4.1 dargestellt. In einem ersten Vorverarbeitungsschritt wird

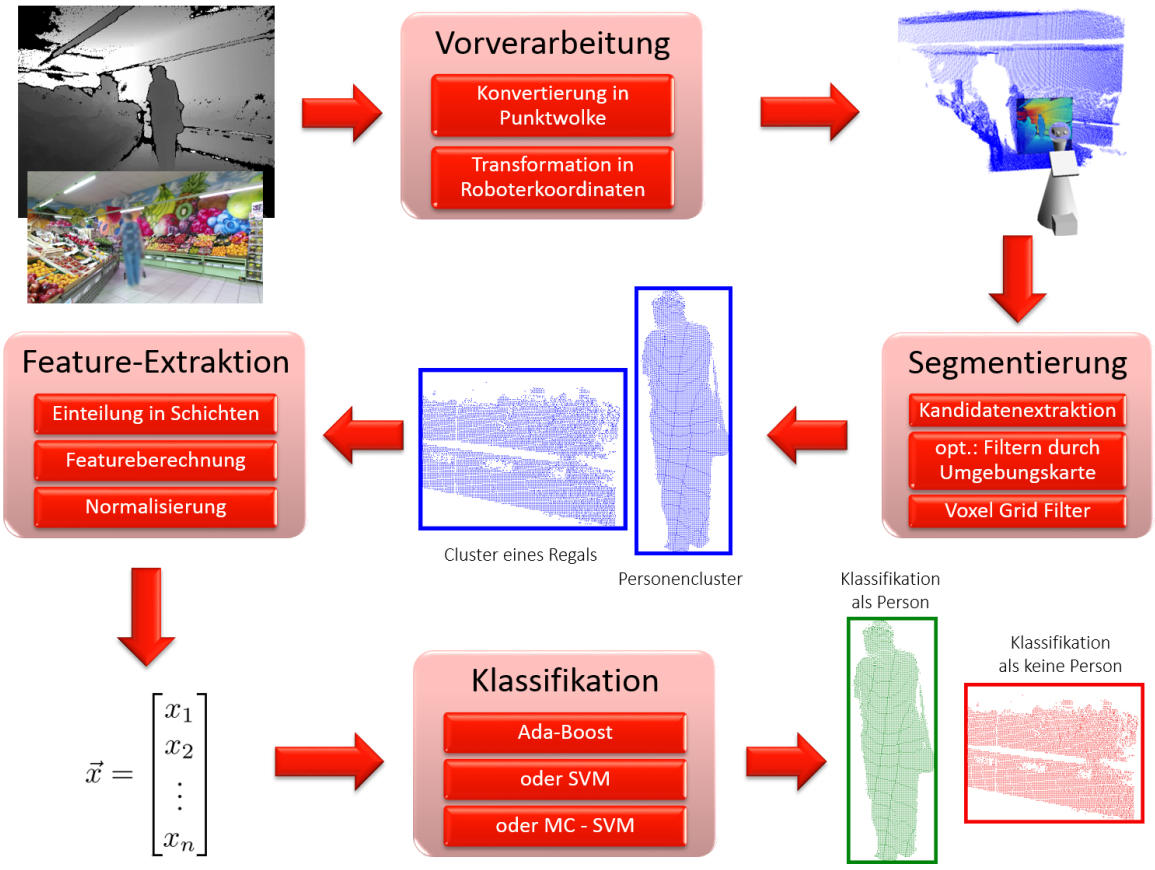


Abbildung 4.1: Architektur des SuPer-Detektors zur Personendetektion in 3D-Punktwolken in einem Supermarkt

das Tiefenbild der Kinect2 in eine 3D-Punktwolke umgewandelt und in Roboterkoordinaten transformiert. Die Segmentierung besteht im eigenen Ansatz aus mehreren Teilschritten. Zunächst werden Kandidatencluster mittels des BlobExtraction-Verfahrens (siehe Abschnitt 4.2.3) extrahiert. Optional findet im Anschluss eine Filterung der Kandidatencluster durch Verwendung der Umgebungskarte (siehe Abschnitt 4.2.4) des Roboters statt, welches zu einer höheren Bildrate zur Anwendungszeit führt. Aufgrund der hohen möglichen Distanz von Personen und der daraus resultierenden sehr geringen Punktdichte in größeren Entfernungen wird jedes Kandidatencluster mittels eines Voxel Grids (siehe Abschnitt 4.2.5) gefiltert. Jedes Cluster wird nach der Segmentierung in den Folgeschritten unabhängig voneinander betrachtet.

4.2 Generierung von Kandidatenclustern

Das Ziel dieses Teilabschnittes ist die Generierung von Kandidatenclustern. Diese Cluster bestehen aus Ausschnitten der Punktwolke, die im optimalen Fall alle zugehörigen Punkte einer Person enthalten oder lediglich Punkte, die zu keiner Person gehören. Jedes Kandidatencluster wird im anschließenden Verlauf einzeln betrachtet und klassifiziert.

4.2.1 Einteilung möglicher Körperhaltungen in Klassen

Im Unterschied zu klassischen Anwendungsfällen der Personendetektion wie z.B. von Fußgängern entsteht im vorliegenden Einkaufsmarktszenario eine größere Vielfalt an unterschiedlichen Körperhaltungen. Es sind neben aufrecht stehenden Personen nicht nur vor Regalen hockende Posen vorhanden, sondern auch vorgebeugte oder mit ausgestrecktem Arm greifende Personen. Zusätzlich können durch Einkaufswägen weitere Posen entstehen, wenn Personen diese schieben oder be- und entladen. Im Rahmen dieser Masterarbeit wurden für die Evaluation drei unterschiedliche Klassen festgelegt, welche durch typische Beispiele in Abbildung 4.2 dargestellt werden.

4.2.2 Vorverarbeitung

In der Vorverarbeitung wird aus jedem Tiefenbild der Kinect2 mit einer Auflösung von 512×424 Pixeln eine Punktwolke mit bis zu 217.088 Punkten berechnet. Für die Aufnahme von Tiefenbildern wird hierbei der libfreenect2-Treiber verwendet, der eine für die Kinect2 neuartige Berechnung der Tiefenwerte implementiert, die korrekte Messungen bis hin zu 18 Metern ermöglicht. Für Details zur Funktionsweise der Tiefenwertbestimmung sei an dieser Stelle auf die Veröffentlichung von [Lawin et al., 2016] verwiesen. Im Anschluss an die Berechnung der Punktwolke wird diese in Roboterkoordinaten transformiert, um Distanzen von Objekten direkt als Entfernung zum Roboter interpretieren zu können.

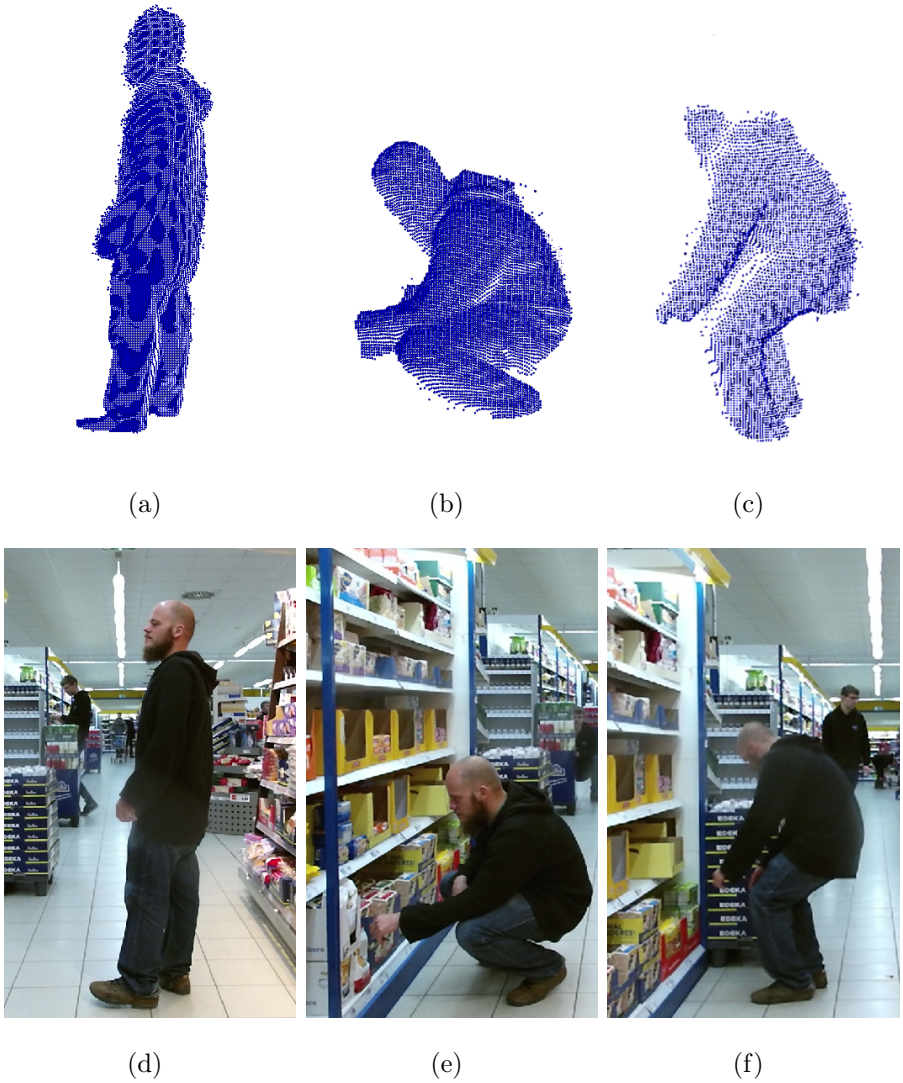


Abbildung 4.2: verwendete Einteilung möglicher Körperhaltungen in drei Klassen *(a) Punktwolkencluster der Klasse* ***stehende Körperhaltung****, (b) Punktwolkencluster der Klasse* ***hockende Körperhaltung****, (c) Punktwolkencluster der Klasse* ***andere Körperhaltung****, (d) Farbbildausschnitt der Klasse* ***stehende Körperhaltung****, (e) Farbbildausschnitt der Klasse* ***hockende Körperhaltung****, (f) Farbbildausschnitt der Klasse* ***andere Körperhaltung***

4.2.3 Segmentierung

Der Segmentierungsschritt bekommt als Eingabe die Punktwolke in Roboterkoordinaten und gibt eine Liste von Clustern aus, die möglicherweise eine Person enthalten. Ein Sliding-Voxel-Ansatz ähnlich zu dem Sliding-Window-Verfahren auf 2D-Bildern ist nicht in einer ausreichenden Rechenzeit möglich. Denn hierbei müsste eine Bounding Box passender Größe an jede Position in der Punktwolke verschoben werden und jedes entstehende Cluster klassifiziert werden. Insbesondere sind in dem Anwendungsszenario *Einkaufsmarkt* sehr viele unterschiedliche mögliche Körperposen gegeben, sodass es eine Vielzahl an möglichen Formen von Bounding Boxen gibt. Jede dieser Boxen müsste bis zu der maximalen Sensordistanz von 18 Metern an jede Position gesetzt werden, welches eine sehr große Anzahl an Kandidatenclustern zur Folge hätte. Zusätzlich befinden sich Personen sowohl sehr nahe an Regalen als auch an anderen Personen. Bei einer Bounding Box fester Größe würden somit bei einer trivialen Segmentierung Teile von Regalen in Clustern mit Personen enthalten sein oder Teile von zwei Personen in einem Cluster. Daher wurde im Rahmen dieser Masterarbeit ein Verfahren ausgewählt, dass in einem Segmentierungsschritt aus der Punktwolke Kandidatencluster extrahiert, die im Anschluss durch einen Klassifikator klassifiziert werden. Die extrahierten Cluster enthalten dabei im Optimalfall lediglich Punkte, die zu einer Klassifikationsklasse gehören.

Blob Extractor

Das Ziel des Verfahrens ist die Generierung von Kandidatenclustern, welche im Anschluss klassifiziert werden. Dabei sollen möglichst viele Bereiche der Punktwolke ausgeschlossen werden, die nicht zu einer Person gehören können, wodurch die Anzahl der zu klassifizierenden Cluster geringer wird. Jeder Punkt der Punktwolke wird hierzu einer der folgenden drei Grundklassen (siehe Abb. 4.3(c)) zugeordnet: *Grundebene*, *Objekte*, *feste Struktur*. Um diese Einteilung zu erreichen, werden die Punkte zunächst in vier verschiedene Schichten eingeteilt (siehe Abb. 4.3(a)):

- **Grundebenen-Schicht**

  Alle Punkte bis zu einer Höhe von 5 Zentimetern werden der Grundklasse *Grun-*

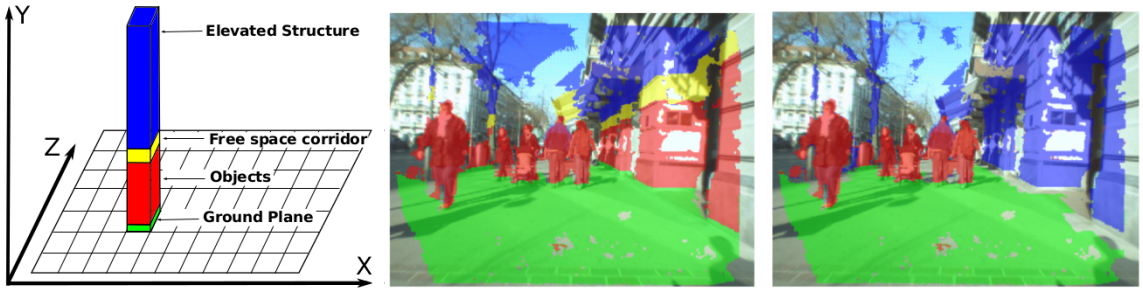
debene zugeordnet und in der Klassifikation nicht weiter berücksichtigt.

- **Objekt-Schicht**
  Alle Punkte die nicht zur Grundebene gehören und niedriger als 2m liegen, werden der Objektschicht zugeordnet. Es gilt die Annahme, dass sich in diesem Höhenbereich der Punktwolke alle Personen befinden.

- **Erhöhte Strukturen**
  Alle Punkte, die höher als 2.30m liegen, gehören direkt zu der Grundklasse *feste Struktur*, da die maximale menschliche Körpergröße überschritten ist. Aufgrund der Annahme, dass sich alle Personen in der Objekt-Schicht befinden, ist diese Schlussfolgerung zulässig. Im öffentlichen Bereich könnten zum Beispiel Personen auf einem Balkon stehen und somit erhöht sein. In dem vorliegenden Anwendungsszenario ist dies jedoch nicht gegeben.

- **Freikorridor**
  Ein naiver Ansatz würde alle Punkte eines Bins zur Grundklasse *feste Struktur* zuordnen, wenn eine hohe Punktdichte in der „Erhöhte Strukturen"-Schicht ab der maximalen Körpergröße gegeben ist. Da in einem Straßenszenario jedoch häufig Gebäudeteile über der Straße hängen, würden Personen unter diesen nicht als *Objekt* klassifiziert werden, sondern als *feste Struktur*. Um dieses Problem zu umgehen, wird zusätzlich der Freikorridor betrachtet. Die Grundannahme des Verfahrens besteht darin, dass sich über den Köpfen von Personen immer ein freier Raum befindet. Aus dieser Annahme heraus wird im Folgenden geschlossen, ob die Punkte in der Objekt-Schicht der Grundklasse *feste Struktur* oder *Objekte* zuzuordnen sind.

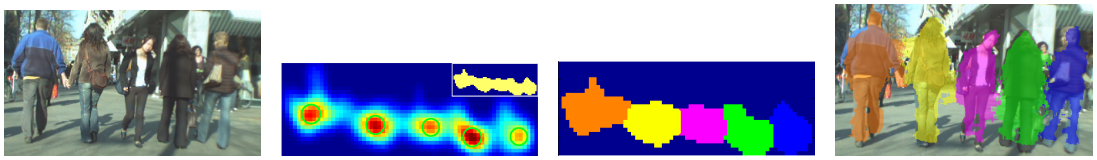
Die Punkte von *festen Strukturen* sind nicht nur in der *„Erhöhte Strukturen"-Schicht* vorhanden, sondern können natürlich sich über mehrere Schichten erstrecken. Dies ist im Szenario *Einkaufsmarkt* sowohl durch Außenwände als auch durch Regale und Kühltruhen gegeben. Die entsprechenden Punkte dieser Objekte sind nach der Einteilung in der *Objekt-Schicht* vorhanden und werden durch die Betrachtung des Freikorridors der Grundklasse *feste Struktur* zugeordnet. Alle Punkte des *Freikorridors* und der Objekt-



(a) (b) (c)

Abbildung 4.3: Einteilung einer Punktwolke nach Strukturen
*(a): Höhenhistogramm der vier Strukturklassen, (b): beispielhafte Einteilung in die vier Strukturklassen, (c): finale Einteilung in die Grundklassen Grundebene, Objekt und feste Struktur, Bilder: [*Jafari *et al., 2014]*

schicht werden jeweils auf ein 2D-Histogramm projiziert. Beinhaltet der *Freikorridor* in einem Bin eine höhere Punktdichte als ein Schwellwert, werden alle Punkte dieses Bins der der Grundklasse *feste Struktur* zugeordnet. Ebenso gehören alle Punkte des entsprechenden Bins der *Objektschicht* zur Grundklasse *feste Struktur*. Besitzt ein Bin des *Freikorridors* nur eine geringe Punktdichte, werden diese Punkte der Grundklasse *feste Struktur zugeordnet*. Alle Punkte des entsprechenden Bins der *Objekt-Schicht* werden der Grundklasse *Objekte* zugeordnet.
Im Folgenden werden nur noch die Punkte der *Objekt-Schicht* sowie das zugehörige Histogramm betrachtet. Um aus der Grundklasse *Objekte* einzelne Kandidatencluster zu extrahieren, werden in dem Histogramm Komponenten ermittelt, die jeweils einem Cluster entsprechen. Hierbei wird zunächst zur Glättung ein Kernel-Filter angewandt. Abbildung 4.4(b) zeigt das entstandene Histogramm einer beispielhaften Szene. Eine einfache Bildung von Zusammenhangskomponenten ist nicht ausreichend, da lediglich eine große zusammenhängende Komponente entstehen würde (siehe Abb. 4.4(b)). Stattdessen wird zur Segmentierung der einzelnen Regionen der *Quick-Shift-Algorithmus* von [Vedaldi und Soatto, 2008] angewandt. Quick-Shift findet die Modale der Dichteverteilung und ordnet jeden Punkt des Histogramms dem nächstgelegenen Maximum zu. Somit werden einzelne Zusammenhangskomponenten im Histo-



(a) (b) (c) (d)

Abbildung 4.4: Extraktion von ROIs
*(a): RGB-Bild, (b): 2D-Histogramm aller Punkte der Grundklasse Objekt mit eingekreisten (grün) lokalen Maxima, Zusammenhangskomponente (rechts oben in gelb) des 2D-Histogramms vor Anwendung des Quick-Shift-Algorithmus, (c): Zusammenhangskomponenten des 2D-Histogramms nach Anwendung des Quick-Shift-Algorithmus, (d): projizierte Punkte aller Bins von jeder zugehörigen Zusammenhangskomponente, Bilder: [*Jafari *et al., 2014]*

gramm der Grundklasse *Objekte* gebildet (siehe Abb. 4.4(d)). Alle zugehörigen Punkte einer Komponente werden zu einem Kandidatencluster zusammengefasst.

Abbildung 4.5(c) zeigt, dass in dem vorliegenden Supermarkt die Regale nicht durchgehend höher als zwei Meter sind. Hierdurch werden die Regale nicht herausgefiltert, da Personen die gleiche Höhe besitzen können. Im Freibereich (gelb markierte Punkte in Abb. 4.5(c)) sind lediglich die Leuchtstoffröhren sowie Teile der Regalhalterung enthalten. Somit wird nur ein geringer Anteil der Regale als feste Struktur (blaue Punkte in Abb. 4.5(d)) definiert und nicht weiter betrachtet. In einem letzten Schritt werden die Bounding Boxen für jedes Cluster bestimmt, dafür wird das Rechteck im projizierten Histogramm berechnet, welches alle dem Cluster zugehörigen Bins umschließt.

Jedes Kandidatencluster wird nun unabhängig voneinander betrachtet und dem Klassifikator übergeben.

4.2.4 Nutzung der Umgebungskarte

Wie in Abschnitt 4.2.3 beschrieben, werden durch die zu verwendenden Schwellwerte viele Kandidatencluster aus der Punktwolke extrahiert, wodurch der Klassifikator im Anschluss sehr oft verwendet werden muss. Um die Anzahl der Kandidatencluster für Personen zu verringern und somit eine Beschleunigung des Detektionssystems, wird

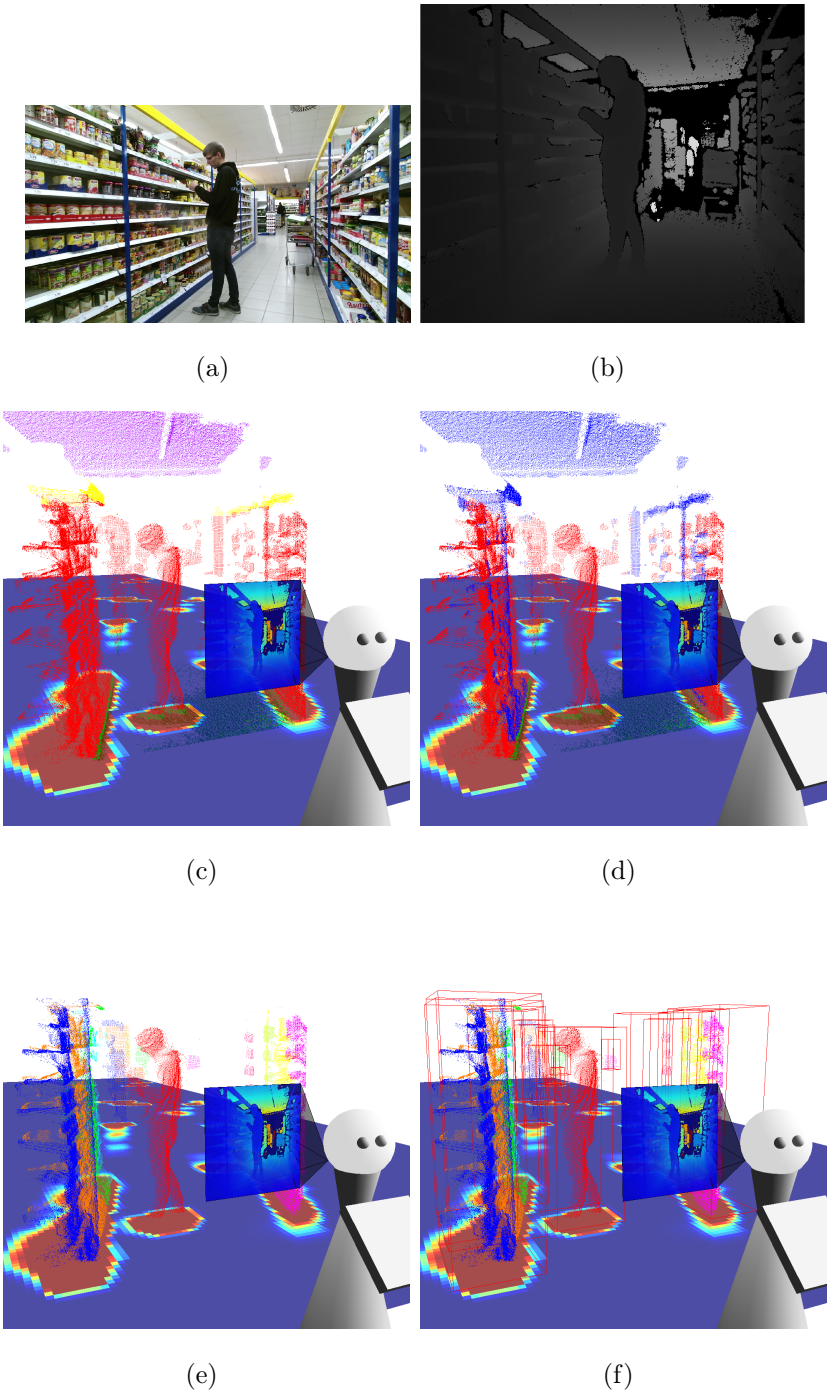


Abbildung 4.5: Kandidatengenerierung mittels BlobExtractor
(a) RGB-Bild, (b) Tiefenbild, (c) Einteilung in Höhenschichten, (d) Finden von festen Strukturen, (e) Kandidatencluster nach Quick-Shift, (f) Kandidatencluster mit Bounding Boxen; Farblegende für (c) + (d): grün: Grundebene, rot: Objektebene, gelb: Freibereich, lila: erhöhte Strukturen, blau: feste Strukturen

nach der Generierung dieser die Lokalisationskarte des Roboters zur Hilfe genommen. Das Ziel dieses Filterungsschrittes ist die Eliminierung von Clustern, die keine Person enthalten können. Während der Fahrt lokalisiert sich der Roboter bereits selbstständig in einer 2D-Karte, um eine Navigation in dem Supermarkt durchführen zu können. Somit kann zu jedem Zeitpunkt die Punktwolke (respektive die Kandidatencluster) an die korrekte Position der Karte gesetzt werden. Liegen Kandidatencluster nun in durch Regale belegten Bereiche, müssen diese nicht im Klassifikationsschritt betrachtet werden. Hierfür werden manuell zwei Belegtheitskarten erstellt. Eine Karte dient der Extraktion von Test- und Trainingsdaten, hierbei werden alle Hindernisse in der Karte eingetragen, die zum Zeitpunkt der Datenaufnahme vorhanden sind (siehe Abb. 4.7(a)). Für die Anwendungsphase wird eine zweite Karte erstellt, in der mögliche Änderungen des Supermarktaufbaus berücksichtigt werden. Durch wechselnde wöchentliche Sonderangebote oder Saisonartikel wie Weihnachtsschokolade werden bestimmte Bereiche des Supermarktes häufig umgebaut. Daher werden für die Anwendungsphase zunächst nur feste Standregale und Kühl- oder Gemüsetheken als Bereiche markiert, in denen zu keinem Zeitpunkt eine Person sich aufhalten kann (siehe Abb. 4.7(a)). Da sich in den markierten Bereichen keine Personen aufhalten können, werden sie *No-Person-Map* genannt.

Selektion der Kandidatencluster

Für jedes aus dem vorigen Schritt entstandene Kandidatencluster muss nun entschieden werden, ob es weiter betrachtet oder verworfen wird. Hierfür wurden zwei verschiedene Ansätze verwendet, welche in der Anwendung ausgewählt werden können. Der erste Ansatz bestimmt den Anteil der Grundfläche der dem Cluster entsprechenden Bounding Box, der in der Hinderniskarte als belegter Raum markiert ist. Liegt der berechnete Anteil über einem festgelegten Schwellwert, wird das Kandidatencluster aus der Menge der zu klassifizierenden Cluster entfernt. Der Vorteil dieses Ansatzes ist die schnelle Berechnung, da lediglich zwei Flächen miteinander verglichen werden. Nachteilig ist jedoch, dass die Bounding Box eine größere Grundfläche besitzt als die Ausdehnung des Clusters, da diese abhängig von der Größe der Bins ist. Somit sind,

aus der Vogelperspektive betrachtet, Bereiche zum Kandidatencluster gehörig, in dem keine Punkte des Clusters sind. Um dieses auszugleichen, wird im zweiten Ansatz nicht die Bounding Box als Grundlage zur Überlappung verwendet. Stattdessen wird jeder Punkt eines Clusters auf die Hinderniskarte projiziert und das Verhältnis der Anzahl der als belegt markierten Pixel mit der Gesamtanzahl der projizierten Pixel berechnet. Im Anschluss erfolgt wie im ersten Ansatz ein Vergleich mit einem festzulegenden Schwellwert. Liegt das Verhältnis über dem Schwellwert, wird das Kandidatencluster aus der Liste der zu klassifizierenden Cluster entfernt und somit angenommen, dass es sich nicht um eine Person handelt.

4.2.5 Voxel Grid Filter

Durch das Lochkameramodell bedingt liegen die Kamerastrahlen in größerer Entfernung weiter auseinander als im Nahbereich. Somit besitzen Kandidatencluster in größerer Entfernung automatisch eine deutlich geringere Punktdichte. Vor der Feature-Extraktion wird ein Voxel Grid Filter angewandt, um in allen Kandidatenclustern eine gleiche Punktdichte zu erhalten. Hierdurch besitzen Personen in unterschiedlichen Entfernungen zum einen eine ähnliche Punktdichte, zum anderen wird die Feature-Extraktion (siehe Abschnitt 4.3) deutlich beschleunigt, da die verwendeten Features für weniger Punkte berechnet werden müssen. Ein Voxel Grid Filter besteht aus einem dreidimensionalen Grid von Voxeln gleicher Größe. Pro Voxel wird ein repräsentativer Punkt aus allen Punkten bestimmt, die in dem Voxel enthalten sind. Alle repräsentativen Punkte bilden zusammen die gefilterte Punktwolke. Für eine geeignete Voxelgröße wurde die Entfernung der Kamerastrahlen in 10m berechnet, da dies der gewünschten Detektionsdistanz entspricht.
Die Kinect2 besitzt einen horizontalen Öffnungswinkel von 70° bei einer Auflösung von 0,1367° und einen vertikalen Öffnungswinkel von von 60° bei einer Auflösung von 0,1415°. In 10 Metern Entfernung beträgt der Abstand der Kamerastrahlen circa 6cm, welcher daher für die Größe der Voxel gewählt wird.

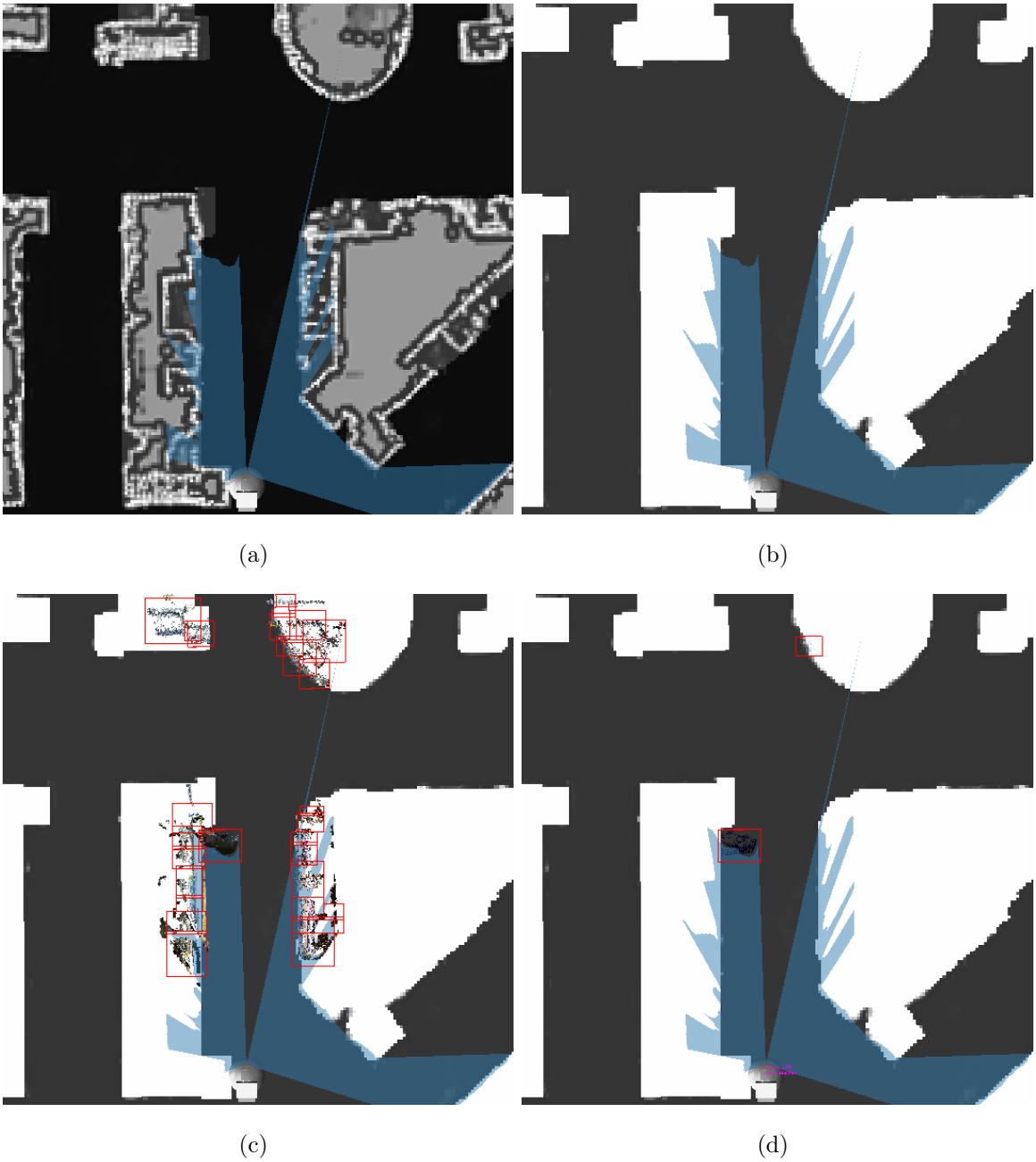
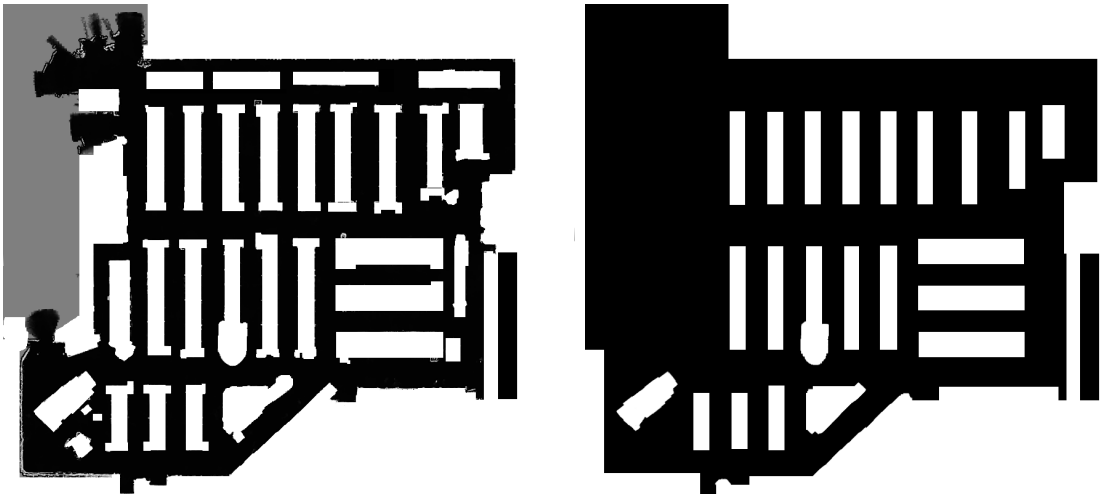


Abbildung 4.6: Verwendung einer No-Person-Map zur Verringerung der Anzahl der Kandidatencluster

(a) Roboterposition in der Lokalisationskarte, (b) lokalisierter Roboter in der No-Person-Map (entspricht den Standorten von Regalen), (c) Kandidatencluster nach Anwendung des Blob-Extractors (siehe 4.2.3), (d) Kandidatencluster nach Filtern durch No-Person-Map; Farblegende: grau: möglicher Aufenthaltsbereich von Personen, weiß: belegter Raum, blau: gültige Laserdistanzmessungen



(a) (b)

Abbildung 4.7: *No-Person-Maps* für die Datenextraktion und die Anwendungsphase

(a) No-Person-Map für die Datenextraktion der Trainings- und Testdaten, (b) No-Person-Map für Anwendungsphase, schwarz: möglicher Aufenthaltsbereich von Personen, weiß: belegter Raum, grau: unbekannter Raum

4.3 Feature-Extraktion

Die in der Segmentierung (siehe Abschnitt 4.2.3) extrahierten Kandidatenclustern werden nun unabhängig voneinander betrachtet. Zur Klassifikation jedes einzelnen Clusters wird aus diesen jeweils ein Featurevektor extrahiert. Hierfür werden im eigenen Ansatz unterschiedliche Möglichkeiten kombiniert. Zum einen findet eine Einteilung in mehrere Schichten statt, zum anderen werden verschiedene Features aus dem jeweiligen Cluster extrahiert.

4.3.1 Einteilung in Schichten

Cluster in Schichten einzuteilen ist ein bewährtes Prinzip, um Verdeckungen von Personen zu behandeln. In [Schneemann, 2013] werden vertikale Schichten verwendet, um gestürzte Personen in einer häuslichen Umgebung zu detektieren. [Martinson und Yalla, 2016] nutzen horizontale Schichten in Tiefenbildausschnitten und erzielten damit gute Ergebnisse (siehe Abschnitt 2.2.2). Im vorliegenden Szenario sind Verdeckungen von Personen sehr häufig, diese treten bereits durch Einkaufswägen auf. Ebenso sind Personen oft nur teilweise sichtbar, weil sie in den engen Gängen von anderen Personen verdeckt werden. Beispielhafte Verdeckungen sind in Abbildung 4.8 ersichtlich. Daher wird im eigenen Ansatz ebenso eine Einteilung der Kandidatencluster in horizontale Schichten verwendet.

4.3.2 Feature-Berechnung

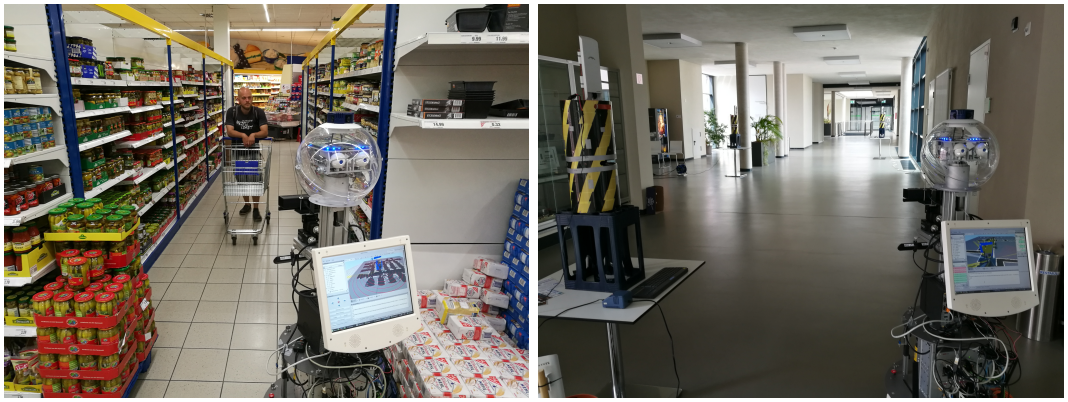
Im Rahmen dieser Masterarbeit werden bekannte Features aus der Domäne der Personen- und Objekterkennung verwendet, die in der Evaluation (siehe Kapitel 5) auf ihre Eignung zur Erkennung von Personen in den szenariospezifischen Körperhaltungen überprüft werden:

- **F**ast **P**oint **F**eature **H**istogram (siehe 3.1.2)
- **V**iewpoint **F**eature **H**istogram (siehe 3.1.3)
- **I**nterest Point Descriptor for **Ro**bust **N**DT-Map Matching (siehe 3.1.4)

Abbildung 4.8: typische Verdeckungen von Personen in einem Supermarkt

- **S**ignature of **H**istograms of **O**rien**t**ations (siehe 3.1.1)

Die Berechnung der Features pro Schicht erfolgt bei dem FPFH, IRON sowie SHOT nach dem gleichen Prinzip. Zunächst wird auf dem gesamten Kandidatencluster das jeweilige Feature für jeden Punkt bestimmt. Bei der Berechnung des Featurevektors eines Punktes können somit auch Punkte einfließen, die sich in einer anliegenden Schicht befinden. Nun wird das arithmetische Mittel der berechneten Features aus allen Punkten einer Schicht berechnet. Die einzelnen Mittelwertsvektoren der Schichten werden aneinander gehängt und bilden den zu klassifizierenden Featurevektor.
Da bei der Berechnung eines VFH das gesamte Cluster betrachtet wird und ein einzelner Featurevektor pro Cluster entsteht und nicht ein Featurevektor pro Punkt des Clusters, findet bei der Verwendung von Schichten ein weiterer Schritt statt. Zunächst werden alle Punkte des Kandidatenclusters in einzelne Subcluster entsprechend der Schichten eingeteilt. Pro Subcluster wird nun ein VFH berechnet und im Anschluss zu einem gesamten Featurevektor verkettet (analog zur Berechnung der anderen Features).



(a) (b)

Abbildung 4.9: Datenaufnahme in verschiedenen Bereichen
(a) Einsatzumgebung des Roboters im Einkaufsmarkt, (b) Datenaufnahme im Zusebau; Bilder: FG NIKR

4.4 Datenaufnahme

Im Rahmen dieser Masterarbeit wurden mehrere Fahrten durch einen Einkaufsmarkt durchgeführt und dabei die notwendigen szenariospezifischen Posen aufgenommen. Zur Generierung von Trainingsdaten wurden zeitlich unabhängige Aufnahmen von mehreren Personen in unterschiedlichen Kombinationen gemacht. Eine typische Situation einer einkaufenden Person mit Einkaufswagen ist in Abb. 4.9(a) ersichtlich. Zusätzlich wurden zur Erhöhung der Diversität an Personen in den Trainingsdaten weitere Aufnahmen im Zusebau durchgeführt (siehe Abb. 4.9(b)). Die aufgenommenen Daten wurden zunächst unter einer strikten Trennung von Ort und Zeit der Aufnahmen in Trainings- und Testdaten zur Evaluierung aufgeteilt.

Extraktion von Trainingsdaten Zur Extraktion von Trainingsdaten wurde vor der Segmentierung 4.2.3 ein weiterer Vorverarbeitungsschritt verwendet. Hierdurch wurde die Anzahl der manuell zu labelnden Kandidatencluster deutlich verringert. Während der Datenaufnahme wurden verschiedene Szenarien durchgespielt, um alle möglichen Körperhaltungen zu den Trainingsdaten hinzufügen zu können. Dieses wurde bei stehendem Roboter durchgeführt, wodurch der Hintergrund (bestehend aus

Regalen des Supermarktes) stets konstant ist. Bei direkter Anwendung der Segmentierung würden aus jedem Bild nahezu dieselben Negativcluster extrahiert werden, welche stets wiederholt gelabelt werden müssten. Daher wurde zunächst über 5 Sekunden ein Hintergrundmodell der Umgebung gelernt. Zur Berechnung des Hintergrundmodells wird das Tiefenbild verwendet. Bei der Segmentierung wird nun von jedem Tiefenbild das Hintergrundmodell abgezogen, wodurch nur noch Punkte überbleiben, die näher an der Kamera sind, als zum Zeitpunkt des Lernens. Das neu berechnete Differenzbild wird nun für die Segmentierung verwendet und beinhaltet lediglich Personen mitsamt ihren Einkaufshilfsmitteln. Durch sensorbedingtes Rauschen werden wie in Abb. 4.10(c) einzelne Pixel fälschlicherweise als Vordergrund bewertet. Diese verrauschten Punkte werden durch den Schritt der Kandidatengenerierung eliminiert, da einzelne Punkte bei der Anwendung des Blob-Extractors (siehe Abschnitt 4.2.3) ein zu geringes Gewicht in ihren jeweiligen Bins erzeugen, um als Kandidatencluster extrahiert zu werden.

4.5 Klassifikation der Featurevektoren

Für die Klassifikation der extrahierten Feature-Vektoren werden im Rahmen dieser Masterarbeit zwei Ansätze verfolgt. Zum einen das Training durch das AdaBoost-Verfahren (siehe Abschnitt 3.2.1) und zum anderen die Klassifikation mittels einer SVM (siehe Abschnitt 3.2.2). In folgendem Kapitel 5 wird ausgewertet, welches Verfahren sich besser für die Detektion von stehenden und hockenden Personen in einem Supermarkt eignet.
Die verwendete Implementierung der SVM führt eine Rastersuche nach den besten Trainingsparametern durch, bei AdaBoost werden verschiedene Parameter durch eigene Experimente verglichen. Anhang B.1 zeigt eine Übersicht über die trainierten Klassifikatoren.

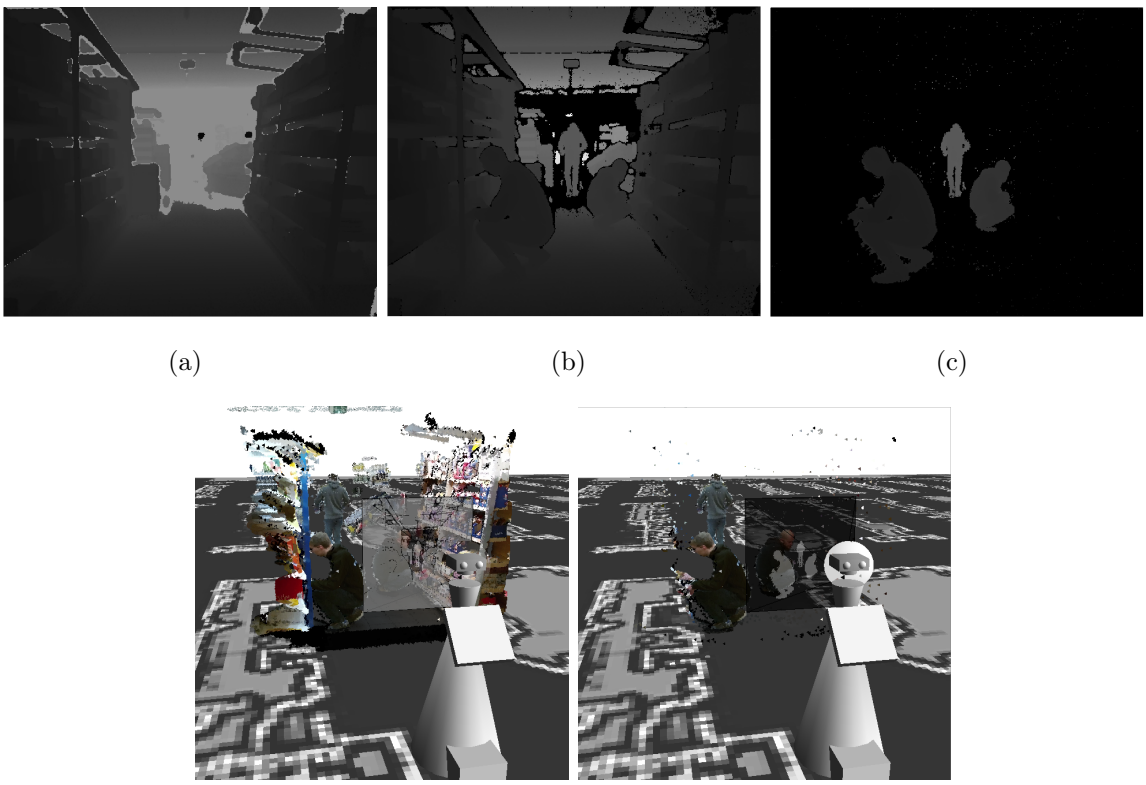


Abbildung 4.10: Extraktion von Vordergrundpixeln durch Hintergrundmodell
(a) gelerntes Hintergrundmodell des Tiefenbildes, (b) Tiefenbild mit allen Pixeln, (c) Tiefenbild mit allen Vordergrundpixeln, (d) Punktwolke aus allen Pixeln des Tiefenbildes, (e) Punktwolke aus allen Vordergrundpixeln des Tiefenbildes

4.6 Zusammenfassung

In diesem Kapitel wurde der SuPer-Detektor zur robusten Erkennung von stehenden und hockenden Personen in einer Einkaufsmarktumgebung vorgestellt. Dieser basiert auf einer anfänglichen Extraktion von Kandidatenclustern aus der 3D-Punktwolke. Aus jedem Cluster werden Deskriptoren berechnet, die eine Abgrenzung von Clustern mit Personen zu Clustern ohne Personen ermöglichen sollen. Als Klassifikator werden im Rahmen dieser Masterarbeit AdaBoost und SVM verwendet. Im folgenden Kapitel 5 werden die einzelnen Ansätze evaluiert und miteinander verglichen, um die beste Kombination der Feature- und Klassifikatorparameter zu ermitteln.

Kapitel 5

Evaluation

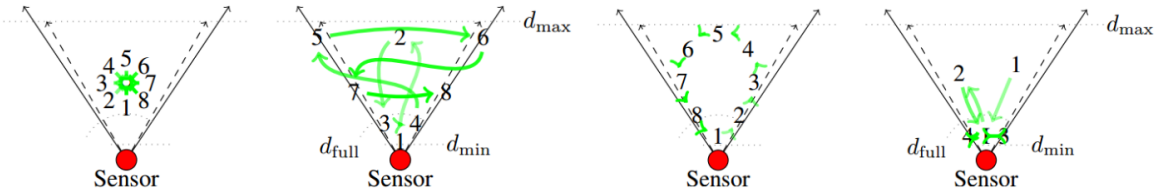
In diesem Kapitel werden die unterschiedlichen Klassifikatoren hinsichtlich ihrer Eignung zur Detektion von stehenden und hockenden Personen in einer Einkaufsmarktumgebung evaluiert. Im Rahmen dieser Masterarbeit wurde untersucht, ob eine Einteilung von Kandidatenclustern in Höhenschichten einen Vorteil für die Personendetektion besitzt. Zudem wurden unterschiedliche Features verwendet, welche in der Objektdetektion häufig verwendet werden. Als Referenzverfahren werden hierbei die in Kapitel 3.3 vorgestellten Detektionssysteme hinzugezogen. Zunächst werden die verwendeten Parameter für die Segmentierung von Kandidatenclustern in der Vorverarbeitung dargestellt. Im Anschluss erfolgt eine Erläuterung der zur Bewertung verwendeten Maße und der Berechnung des Verdeckungsgrades von Personen. Die einzelnen Ergebnisse der Experimente werden vor einer abschließenden Zusammenfassung vorgestellt.

5.1 Aufnahme von Trainings- und Testdatensatz

Zu Beginn der Masterarbeit existierten keine Trainings- oder Testdatensätze, die das breite Spektrum der möglichen Körperhaltungen von Menschen in einer Einkaufsmarktumgebung abdecken. Daher wurden neue Daten aufgenommen, um diese zu erstellen.

5.1.1 Trainingsdatensatz

Hierfür wurden mittels der in Abschnitt 4.2.3 vorgestellten Segmentierung Cluster aus den aufgenommenen Daten extrahiert. In einem neu entworfenen Label-Tool werden diese Kandidatencluster manuell mit Labelinformationen versehen, welche zahlreiche Informationen über jedes Cluster enthalten. Dabei wird für jedes Cluster gespeichert, um welche Klasse (siehe Abschnitt 4.2.1) es sich handelt. Weiterhin wurde die Beschaffenheit des Clusters näher definiert in Hinblick auf Verdeckung und das Beinhalten eines Einkaufswagens. Diese Informationen sind unter anderem für die zukünftige Verwendung der Daten potentiell nützlich und wurden teilweise im Rahmen dieser Masterarbeit nicht verwendet.
Der Trainingsdatensatz beinhaltet aus eigenen Datenaufnahmen 5600 Punktwolkencluster stehender Personen, 1150 Cluster mit nicht stehenden Personen sowie 55.005 Negativbeispiele. Dabei wurden der Großteil der Cluster der Negativbeispiele aus Durchfahrten im Supermarkt extrahiert, in denen keine Personen enthalten waren. Die Personencluster wurden manuell mittels des eigens entworfenen Label-Tools klassifiziert.
Da die selbst durchgeführten Aufnahmen nur einen kleinen Personenkreis enthalten, wurde zusätzlich ein Teil des SRL-Datensatzes (siehe [Linder et al., 2015]) zum Training verwendet. Darin enthalten sind Personen unterschiedlichen Alters und Geschlechts, welche in vielen Orientierungen und Entfernungen aufgenommen wurden. Abbildung 5.1(a) zeigt die vier verschiedenen Bewegungsabläufe der Testpersonen. Durch Einbezug eines Teils des Datensatzes wird die Diversität der Trainingsdaten deutlich erhöht und somit eine unter Umständen bessere Generalisierung erreicht.
Aufgrund der Verwendung der in Abschnitt 4.10 beschriebenen Vordergrundsegmentierung sind im Training nur Personen bis zu einer Entfernung von circa 12 Metern enthalten.



(a)

(b)

Abbildung 5.1: Inhalt des SRL-Datensatzes von [Linder et al., 2015]
(a) 4 Bewegungssequenzen der aufgenommenen Personen in unterschiedlichen Orientierungen (Zahlen: abzulaufende Positionen, d_{max}: maximale Entfernung einer Person, d_{min}: minimale Entfernung einer Person, d_{full}: Entfernung, bei der die Personen vollständig sichtbar sind), (b) Beispielausschnitte von Personen aus dem Datensatz, Bilder entnommen aus [Linder et al., 2015]

5.1.2 Erstellung des SuPer-Datensatzes

Für den Testdatensatz wurden Aufnahmen verwendet, die nicht im Trainingsdatensatz enthalten sind. Wie zuvor beschrieben werden die Kandidatencluster segmentiert und in die verschiedenen Klassen eingeteilt. Die extrahierten 3D-Bounding-Boxen werden auf das Farb- und Tiefenbild projiziert, wodurch ein Vergleich mit Referenzverfahren ermöglicht wird, die nur 2D-Detektionsboxen bestimmen. Zum einen ist jedoch nicht gesichert, dass alle Personen korrekt in Cluster segmentiert wurden. Zum anderen besitzen die Farb- und Tiefenkamera der Kinect2 einen unterschiedlichen Sichtbereich, wodurch Personen teilweise nur im Farbbild (rechts und links am Rand) sichtbar sind. Des Weiteren sind bei der Segmentierung unter Umständen zum Beispiel ausgestreckte Hände nicht in dem Cluster der Person enthalten, da die Punktdichte in der Objektschicht in dem Bereich nicht ausreichend hoch ist (die Erläuterung der Schichten ist in Abschnitt 4.2.3 erläutert). Diese Fälle müssen in einem weiteren Schritt behandelt werden, um einen korrekt gelabelten Datensatz zu erhalten. Daher wird die Größe der projizierten 2D-Bounding-Boxen manuell korrigiert. Personen, die nur im Farbbild sichtbar sind, wurden für den SuPer-Datensatz gelabelt, werden aber in der Evaluation für diese Masterarbeit nicht betrachtet. Diese Personen könnten durch kein Detektionsverfahren auf dem Tiefenbild erkannt werden und würden daher Verfahren auf dem Farbbild bevorzugen.

Der zur Evaluation verwendete Testdatensatz enthält somit alle Informationen über Personen sowohl in 2D und 3D. Tabelle 5.1 zeigt die Aufteilung der 4303 gelabelten Personen in unterschiedlichen Körperhaltungen. Es findet eine Unterscheidung statt zwischen Aufnahmen während der Fahrt des Roboters und dem Stand des Roboters. Um alle möglichen Körperhaltungen in unterschiedlichen Entfernungen im Szenario Einkaufsmarkt zu erhalten, wurden bei stehendem Roboter typische Situationen nachgestellt. Hieraus resultierend sind in jedem Bild nahezu dieselben negativen Beispiele enthalten. Klassifiziert ein Detektionssystem eines dieser Beispiele falsch positiv, würde dies in jedem Bild zu einer falsch positiven Detektion führen und die Auswertung somit verfälschen. Daher werden bei der Evaluation auf dem SuPer-Datensatz falsch positive Detektionen nur verwendet, wenn der Roboter sich in der Bewegung befin-

	0m - 7m	**7m - 10m**	**10m - 18m**	**gesamt**
stehend	2023	578	373	2974
hockend	580	127	0	707
andere	374	186	62	622
gesamt	2977	891	435	4303

Tabelle 5.1: Anzahl Personen im SuPer-Datensatz

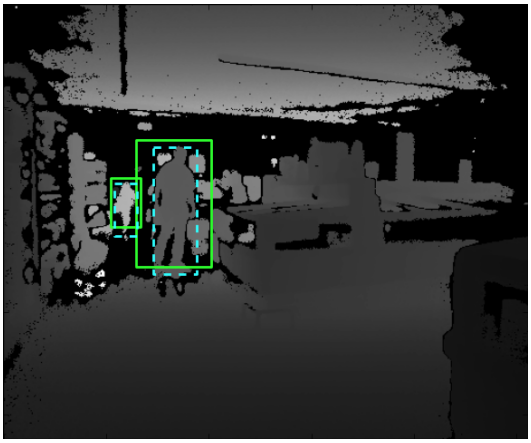
Die Distanzangaben beziehen sich auf die Entfernung zum Sensor.

det. In Anhang in Tabelle A.2 ist die Anzahl der Personen bei stehendem Roboter dargestellt. Bei Bildern, die während der Fahrt entstanden sind, werden alle Detektionen gewertet. Eine Übersicht der zu detektierenden Personen ist ebenso im Anhang in Tabelle A.1 ersichtlich.

5.2 Bewertungsmaß zur Evaluation

Als Maß für die Güte der Klassifikatoren wird im Rahmen dieser Masterarbeit die *Detection Error Tradeoff Curve* verwendet. Auf der x-Achse wird die Anzahl der falsch positiven Detektionen pro Bild abgetragen und auf der y-Achse die Miss Rate. Die Miss Rate bezeichnet den Anteil der Personen, die in einem Bild nicht korrekt klassifiziert werden. Dieses Maß bewertet die Güte des gesamten Detektionssystems und nicht nur von einzelnen Ausschnitten wie bei der Verwendung der False Positives Per Window. Die Auswertung der Detektionen erfolgt auf den zweidimensionalen Farb- und Tiefenbildern, um einen Vergleich mit anderen Detektionsverfahren zu ermöglichen, die lediglich 2D-Detektionsboxen berechnen. Wie [Dollar et al., 2012] in ihrer Ausarbeitung darstellen, wird auch in dieser Arbeit zur Entscheidung der Übereinstimmung einer Ground-Truth-Box mit einer Detektionsbox das Verhältnis des Schnittes zur Vereinigung der Boxen bestimmt (siehe Gleichung 5.2(a)). Ab einem Schwellwert von 50% werden die zwei Boxen als zueinander zugehörig gewertet.

$$a_o = \frac{area(BB_{dt} \cap BB_{gt})}{area(BB_{dt} \cup BB_{gt})}$$



(a) (b)

Abbildung 5.2: Berechnung der Überlappung einer Detektionsbox mit einer Ground-Truth-Box
*(a) Formel zur Überlappung, (b) Beispielmatching; nach [*Dollar *et al., 2012], a_o: area of overlap, BB_{dt}: Detektionsbox des Klassifikators, BB_{gt}: Ground-Truth-Box, türkis: Ground Truth, hellgrün: gematchte Detektion*

5.3 Verwendete Klassifikatoren

Im Rahmen dieser Masterarbeit wurden unterschiedliche Klassifikatoren zur Personendetektion trainiert. Zunächst wurde das von [Freund und Schapire, 1997] vorgestellte AdaBoost-Verfahren gewählt, da es insbesondere nur relativ kurze Trainingszeiten benötigt. Die vorhandene Implementierung der OpenCV-Bibliothek (siehe [opencv, 2017]) unterstützt das Training und die Anwendung für Zweiklassenprobleme. Daher wurden die in Abschnitt 4.2.1 vorgestellten drei Klassen *stehende Körperhaltung*, *hockende Körperhaltung* und *andere Körperhaltung* zu einer Positivklasse zusammengefasst. Das Training einer SVM (siehe 3.2.2) benötigt einen langen Zeitraum, daher erfolgt im Rahmen dieser Arbeit eine Auswertung auf dieser Basis nur für wenige Parameter und Features.

5.3.1 Parametrisierung der Segmentierung und Feature-Extraktion

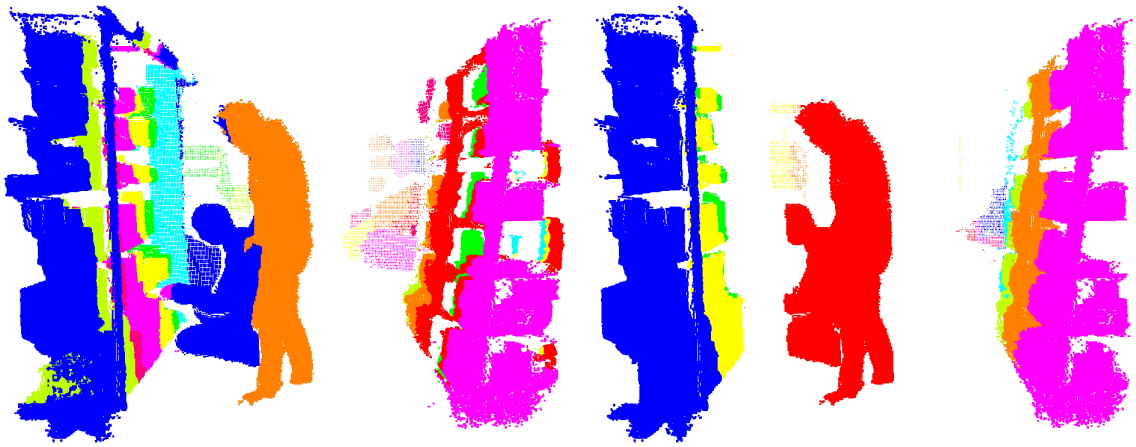
In den Schritten bis zur Bildung der Feature-Vektoren sind zahlreiche Parameter erforderlich, um eine möglichst robuste Personendetektion zu ermöglichen.

Segmentierung

Die Größe der extrahierten Cluster durch das Verfahren des *Blob Extractor* (siehe Abschnitt 4.2.3) ist maßgeblich von der Parametrisierung des Histogramms, welches durch die Projektion der Objektschicht auf die Bodenebene gebildet wird, sowie des darauf angewandten Kernelfilters abhängig. Im Rahmen dieser Masterarbeit wurde evaluiert, welche Größe der projizierten Bins optimal ist. Bei größeren Bins fallen Teile von Regalen in dieselben Bins, die Personen zugehörig sind, wodurch der Klassifikationsschritt erschwert werden würde. Werden die Bins zu klein gewählt, können bei der Projektion einer einzigen Person auf die Grundfläche mehrere Maxima im Histogramm entstehen, hierdurch würde die Person in mehrere Cluster geteilt werden. Durch die Anwendung eines Gaußfilters wird das Histogramm geglättet, was der Aufteilung entgegenwirkt. Die Bedeutung der einzelnen Parameter ist in Abschnitt 4.2.3 erläutert.

In dem vorliegenden Einsatzszenario wurden für die Anwendung folgende Parameter qualitativ bestimmt (X-Richtung entspricht der Fahrtrichtung, Y-Richtung der seitlichen Verschiebung):

- min. Distanz X-Richtung: 50cm
- Bingröße X-Richtung: 3cm
- Bingröße Y-Richtung: 3cm
- Kernelgröße X-Richtung 7cm
- Kernelgröße Y-Richtung 7cm
- Gewichtsschwellwert der Objekt-Schicht: 200



(a) (b)

Abbildung 5.3: Vergleich der extrahierten Cluster durch verschiedene Parameter des *Blob Extractor*

Die einzelnen Cluster sind farblich voneinander getrennt dargestellt, (a) im Rahmen dieser Masterarbeit verwendete Parametereinstellung, (b) extrahierte Cluster bei der Wahl von größeren Bins

Eine detaillierte Evaluierung der Parameter wurde im Rahmen dieser Masterarbeit nicht durchgeführt, da es sich um eine Vorarbeit handelt. Abbildung 5.3(a) zeigt beispielhafte Bilder der Clusterbildung unter den gewählten Parametern. Dazu sind in Abbildung 5.3(b) die Ergebnisse der Clusterbildung dargestellt, wenn z.B. die Bins größer gewählt werden.

Feature-Extraktion

Alle verwendeten Deskriptoren verwenden die Oberflächennormalen der Punkte in einem Kandidatencluster, welche über eine Hauptkomponentenanalyse, wie in [Rusu, 2009] beschrieben, bestimmt werden. Bei der Berechnung der Normale eines Punktes fließen mehrere benachbarte Punkte aus einer Punktwolke ein. Für die Bestimmung der benachbarten Punkte gibt es zwei Möglichkeiten: *Nächste-Nachbarn-Suche (kNN)* und *Radius-Suche*. *kNN* bezeichnet die Suche nach den k nächsten Nachbarn bezogen auf die euklidische Distanz. Bei der Radius-Suche hingegen gelten alle Punkte als benachbart, dessen Distanz zum Anfragepunkt geringer als der Radius ist. In 3D entspricht dies einer kugelförmigen Nachbarschaft. Der Vorteil der Radius-Suche besteht darin, dass lediglich Punkte mit in die Normalenberechnung eingehen, die sich auch in der Nähe des Punktes der zu berechnenden Normale befinden. Bei dem kNN-Verfahrens können bei einer spärlichen Punktwolke auch Punkte in der Nachbarschaft enthalten sein, die weit entfernt voneinander liegen. Jedoch ist die Berechnung der k nächsten Nachbarn berechnungseffizienter als die Suche nach allen Punkten in einem Radius. In dieser Masterarbeit wird daher zunächst die Radiussuche verwendet und der beste Klassifikator im Anschluss verglichen mit der Nutzung des kNN-Verfahrens.

Ein weiterer untersuchter Einflussfaktor ist die Verwendung unterschiedlicher Suchoberflächen bei der Suche nach den Nachbarn zur Berechnung der Oberflächennormalen. Die Features werden wie in Abbildung 4.1 dargestellt für alle Punkte einer gefilterten Punktwolke (siehe Abschnitt 4.2.5) berechnet. Die Bestimmung der benachbarten Punkte für die Normalenberechnung kann unter Verwendung dieser gefilterten Punktwolke geschehen oder es wird die originale Punktwolke verwendet. Werden die Normalen auf der ursprünglichen Punktwolke berechnet, benötigt dies mehr Rechenzeit bei der Radiussuche (da deutlich mehr Punkte in dem gleichen Radius liegen). Die Norma-

len entsprechen jedoch genauer der natürlichen Oberfläche, da durch die Anwendung des Voxel-Grid-Filters Punkte und somit Informationen verworfen werden. Die Auswirkung der Suchoberfläche wird ebenso anhand des besten Klassifikators evaluiert. Zunächst werden die benachbarten Punkte für die Berechnung der Oberflächennormalen auf der ursprünglichen, nicht gefilterten Punktwolke berechnet.
Wie in Kapitel 4 erläutert, findet die Feature-Extraktion pro Kandidatencluster statt. Mit Ausnahme des VFH (siehe Abschnitt 3.1.3) werden die verwendeten Deskriptoren pro Punkt berechnet und nicht pro Cluster. Bei der Feature-Extraktion werden die einzelnen Deskriptoren zu einem Deskriptor pro Schicht zusammengefasst, indem das arithmetische Mittel aller Deskriptoren der zu einer Schicht gehörenden Punkte berechnet wird. Bei der Verwendung von mehreren Schichten werden die Mittelwert-Deskriptoren von jeder Schicht aneinandergehängt und ergeben zusammen den zu klassifizierenden Feature-Vektor.

5.3.2 AdaBoost-Klassifikatoren

Im Rahmen dieser Masterarbeit wurden verschiedene Featureparameter mit verschiedenen AdaBoost-Parametern kombiniert. In diesem Abschnitt wird ein Überblick über die getesteten Parameter geschaffen, in der Evaluation wird pro Feature näher auf die Veränderungen durch diese Varianten eingegangen.
Mittels des AdaBoost-Verfahrens (siehe 3.2.1) wurden verschiedene Klassifikatoren trainiert. Hierbei fanden das FPFH (siehe Abschnitt 3.1.2), IRON-Features (siehe Abschnitt 3.1.4), der SHOT-Deskriptor (siehe Abschnitt 3.1.1) sowie das VFH (siehe Abschnitt 3.1.3) Anwendung. Für jedes Feature wurden dabei AdaBoost-Klassifikatoren mit unterschiedlichen Parametern trainiert, um den besten Klassifikator zu ermitteln. Zunächst wurde mit 100 Decision Trees der maximalen Tiefe 3 als Weak Learner trainiert. Diese Kombination erzielte bereits gute Ergebnisse, im Weiteren wurden zum Vergleich dieselben Trainingsdaten verwendet mit 500 Decision Trees der maximalen Tiefe 5.
Eine weitere Parametervariante entsteht durch die geringe Punktdichte der Punktwolke in größeren Entfernungen. Je weiter entfernt Objekte sich befinden, desto weniger

Kamerastrahlen treffen auf sie. Somit enthält die 3D-Punktwolke in größeren Entfernungen deutlich weniger Punkte und voraussichtlich eine große Streuung der Features. Daher wurde im Anschluss die beste Kombination der AdaBoost-Parameter verwendet, um einen Klassifikator zu mit Trainingsdaten zu trainieren, die in maximal 7 Metern Entfernung liegen.
Tabelle B.1 zeigt die trainierten Klassifikatoren unter Verwendung des FPFH, Tabelle B.2 die Klassifikatoren für IRON-Features, Tabelle B.3 bei Nutzung des SHOT-Deskriptors sowie in Tabelle B.4 für das VFH.

5.3.3 SVM-Klassifikatoren

Wie in Abschnitt 3.2.2 dargestellt, existieren mehrere Varianten, eine SVM zur Klassifikation zu verwenden. Zunächst wurden lineare SVMs trainiert. In Test zeigte sich jedoch sehr deutlich, dass eine Nutzung von linearen SVMs nicht sinnvoll ist, da sehr viele Falschdetektionen auftraten. Daher lässt sich annehmen, dass die vorliegenden Trainingsdaten nicht linear separierbar sind und daher zur Trennung durch eine Hyperebene in einen höherdimensionaleren Raum transformiert werden müssen (siehe Abb. 3.5). Als Kernelfunktion wurde der RBF-Kernel (*Radial Basis Function*) verwendet.
Die Trainingszeiten der SVMs sind erheblich länger, wodurch im Rahmen dieser Masterarbeit exemplarisch Ergebnisse von SVMs gezeigt werden, die zu einem früheren Zeitpunkt der Bearbeitungszeit trainiert wurden. Aus den Ergebnissen wird abgeleitet, ob eine nähere Untersuchung der Klassifikation mit einer SVM sinnvoll ist und potentiell zu einer besseren Detektionsleistung führt.
Zusätzlich wird evaluiert, ob nicht nur eine reine Personendetektion möglich ist, sondern auch eine Klassifikation der Körperpose. Hierfür wurden mit verschiedenen Features Multi-Klassen-SVMs (siehe Abschnitt 3.2.3) trainiert, welche die Testdaten als negativ, nicht stehende oder stehende Person klassifizieren. Bei den Trainingsdaten wurden zu diesem Zweck die Klassen *hockende Körperhaltung* und *andere Körperhaltung* zu einer Klasse zusammengefasst. Die Definitionen der einzelnen Körperhaltung ist in Abbildung 4.2 dargestellt. Die OpenCV2-Implementierung der Multi-Klassen-SVM besitzt als Output nur die Klassenentscheidung und keine Konfidenz. Daher

können für diese Klassifikatoren im *Detection-Error-Tradeoff-Diagramm* (Erläuterung des Diagramms in Abschnitt 5.2) keine Kurven erstellt werden, da die Klassenentscheidung nicht anhand eines Schwellwertes getroffen wird. In der Evaluation wird daher nur ein Punkt als Vergleich zu den Kurven der anderen Klassifikatoren dargestellt.

5.4 Berechnung des Verdeckungsgrades

Verdeckte Personen zu detektieren ist selbstverständlich schwerer als eine vollständig sichtbare Person. Daher wird in der Evaluation zusätzlich betrachtet, ob der eigene Ansatz verdeckte Personen besser detektieren kann als andere Verfahren. Für die Berechnung des Verdeckungsgrades wird ein Rechteck um die Projektion eines Clusters auf die Bildfläche gebildet (siehe Abb. 5.4). Jeder Tiefenwert eines Pixels des Rechtecks wird mit dem entsprechenden Pixel im ursprünglichen Tiefenbild verglichen. Besitzen beide Pixel den Wert 0, gab es an dieser Stelle keine gültige Messung. Ist der Tiefenwert größer als die hintere Seite der 3D-Bounding-Box, wird dieser Pixel dem Hintergrund zugeordnet. Alle Punkte, die den gleichen Tiefenwert in beiden Ausschnitten haben, gehören zu dem Cluster selbst. Punkte, die in der Projektion einen geringeren Tiefenwert (Bild für Projektion wird mit Nullwerten initialisiert) besitzen als im ursprünglichen Tiefenbild, werden durch etwas verdeckt. Der Verdeckungsgrad berechnet sich nun aus dem Anteil der verdeckten Pixel an der Gesamtanzahl der Pixel (siehe Gleichung 5.1).

$$Occ(roi) = \frac{\#(verdecktePixel)}{\#Reihen \cdot \#Spalten} \tag{5.1}$$

Der Testdatensatz wird in vier Klassen je nach Grad der Verdeckung eingeteilt (siehe Tabelle 5.2). Diese Einteilung wurde durch [Dollar et al., 2012] vorgestellt und in dieser Masterarbeit ebenso verwendet, um vergleichbare Ergebnisse zu erhalten.

(a) (b) (c) (d)

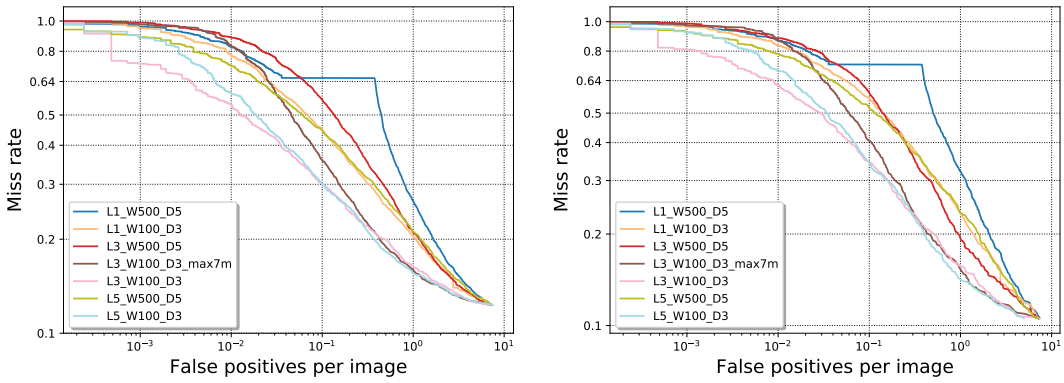
Abbildung 5.4: Berechnung des Verdeckungsgrades
(a) ROI im Farbbild, (b) ROI im Tiefenbild, (c) Projektion des Clusters auf Bild, (d) Einteilung in Vorder- und Hintergrund, grün: Vordergrundpixel, orange: verdeckte Pixel, grau: Hintergrundpixel

Verdeckungsanteil [in %]	**Klasse**
0	none
1 - 35	partial
36 - 80	heavy
>80	full

Tabelle 5.2: Einteilung der Ground-Truth-Daten in Verdeckungsklassen nach [Dollar et al., 2012]

5.5 Evaluation

In diesem Abschnitt werden die Ergebnisse der Klassifikatoren auf dem SuPer-Datensatz verglichen. Zunächst findet eine Auswertung der trainierten AdaBoost-Klassifikatoren statt. Dabei wird pro verwendetes Feature untersucht, welche Parameter der vorgestellten Möglichkeiten (unterschiedliche Anzahl an Schichten, unterschiedliche Anzahl an Weak Learnern, maximale Trainingsdistanz, Entropieschwellwert des IRON-Deskriptors) optimal für die Personendetektion in einer Einkaufsmarktumgebung unter Verwendung von 3D-Punktwolken sind. Die jeweils besten Klassifikatoren werden im Anschluss miteinander verglichen, woraus der im Rahmen dieser Masterarbeit beste AdaBoost-Klassifikator resultiert. Anhand exemplarischer 2-Klassen-SVMs findet im Weiteren eine Bewertung statt, ob die Verwendung von SVMs einen Vorteil bezüglich der Detektionsgüte gegenüber AdaBoost-Klassifikatoren besitzt.
Der Abschnitt schließt mit dem Vergleich der besten eigenen Klassifikatoren und der in Abschnitt 3.3 vorgestellten Referenzverfahren.
Im Rahmen dieser Evaluation wird zum Vergleich von zwei Klassifikatoren jeweils die Miss Rate bezogen auf 0.1 False Positives Per Image betrachtet. Die Einteilung in verschiedene Evaluationsklassen findet analog zu den Definitionen von [Dollar et al., 2012] statt. In der Evaluationsklasse *reasonable* sind alle Personen mit einer minimalen Höhe von 50 Pixeln sowie keiner oder teilweisen Verdeckung enthalten. Die Auswertung auf den *occluded* Beispielen beinhaltet ausschließlich Personen, welche verdeckt werden. Im Anhang dieser Masterarbeit werden jeweils die Kurven der Evaluationsklasse *overall* dargestellt, welche alle zu detektierenden Personen enthält. Da die Aufgabenstellung eine Detektion von Personen bis zu 10 Metern vorsieht, wird die Auswertung im Rahmen dieses Kapitels bis zu dieser Distanz ausgeführt. Die DET-Kurven für weitere Distanzen sind im Anhang dargestellt.



(a) (b)

Abbildung 5.5: DET-Kurve der AdaBoost-Klassifikatoren unter Verwendung des SHOT

(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume

5.5.1 Evaluation der AdaBoost-Klassifikatoren

SHOT zur Personendetektion

Abbildung 5.5 zeigt die DET-Kurven für die Verwendung des SHOT-Features. Die Einteilung der zu klassifizierenden Cluster in Schichten führt zu einer deutlichen Verbesserung der Klassifikationsgüte. Zwischen der Einteilung in drei und fünf Schichten ergibt sich jedoch kein signifikanter Unterschied. Da die Histogramme des SHOT-Features aus 352 Bins besteht, fallen bei einer Einteilung in mehr Schichten deutlich weniger Punkte in einzelne Bins. Dies kann dazu führen, dass bei mehr Schichten eine zu spärliche Besetzung der Histogramme entsteht. Bei einer Begrenzung der maximalen Distanz eines Clusters im Training des Klassifikators auf 7 Metern, führt dies zu einer Erhöhung der Miss Rate um 5,8% und somit zu einer Verschlechterung.
Die Verwendung von mehr Weak Learnern mit einer größeren Tiefe führt zu einem deutlich schlechteren Ergebnis, da unter Umständen der Trainingsdatensatz zu klein für die Trennung mit 500 Weak Learnern und einer höheren maximalen Baumtiefe ist.

IRON zur Personendetektion

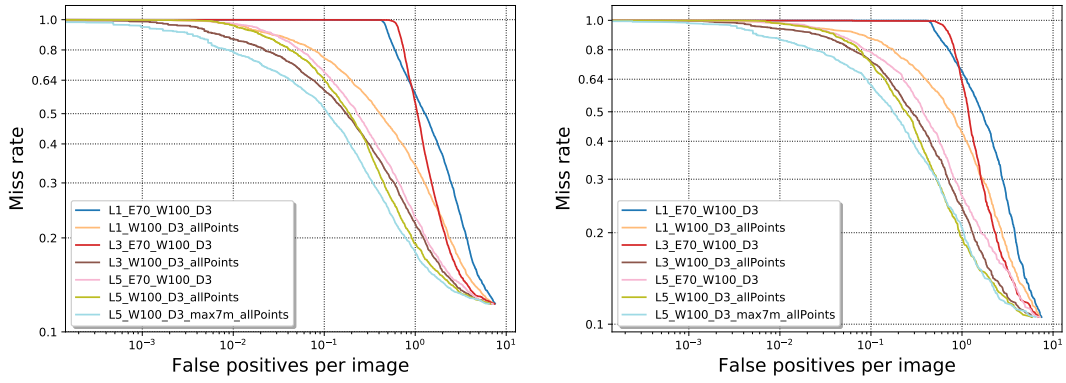
Bei der Verwendung von IRON-Features kann ein Schwellwert für die minimale Entropie eines Deskriptors verwendet werden (siehe Abschnitt 3.3). Liegt die Entropie unterhalb des Schwellwertes, fließen die Histogramme nicht ein bei der Bildung der Featurevektoren pro Schicht. Der Einfluss dieses Schwellwertes wird im Folgenden untersucht. Die Evaluation der Parameter für das Training des AdaBoost-Verfahrens mit 500 Weak Learnern ist in Anhang B.4 ersichtlich.
Abbildung 5.6 zeigt, dass die Einteilung in mehr als eine Schicht auch bei der Verwendung der IRON-Features zu einer robusteren Personendetektion führt. Werden alle IRON-Deskriptoren verwendet und nicht nur solche mit einer Entropie höher als 0.7, nimmt die Detektionsleistung des Systems deutlich zu. Eine Beschränkung der Trainingsdaten auf 7 Metern führt zu einer weiteren Verbesserung. Insgesamt ergibt sich jedoch bei der Verwendung von IRON-Features eine schlechte Detektionsleistung, welche durch mehrere Faktoren begründet sein kann. Die Mittelwertbildung aller IRON-Deskriptoren pro Schicht könnte wichtige Informationen der Punkte mit einer sehr hohen Entropie verringern. Da die Beschränkung der Trainingsdaten auf 7 Meter eine deutliche Verbesserung bewirkt, könnte der Radius der Nachbarschaft bei der Berechnung der IRON-Deskriptoren zu groß sein.

FPFH zur Personendetektion

Bei der Verwendung des FPFH ergeben sich die in Abb. 5.7 dargestellten DET-Kurven. Da sich bei der Einteilung in fünf statt drei Schichten bei den *reasonable* Personen eine Verbesserung um 0.085 ergibt, wurde für dieses Feature überprüft, ob eine Einteilung in sieben Schichten die Klassifikationsergebnisse weiter verbessert. Da sich hierbei eine Verschlechterung der Miss Rate von 0,55% ergibt, führt eine Einteilung in mehr als fünf Schichten zu keiner weiteren Verbesserung.

VFH zur Personendetektion

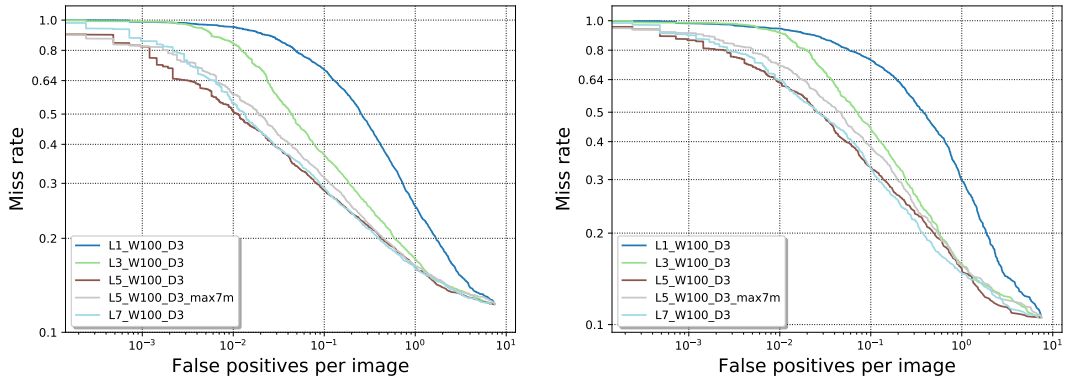
Abbildung 5.8 zeigt die Detektionsleistung bei Verwendung des VFH als Deskriptor. Die Evaluation erfolgt im Rahmen dieser Masterarbeit nur für die Anwendung mit



(a) (b)

Abbildung 5.6: DET-Kurve der AdaBoost-Klassifikatoren unter Verwendung des IRON

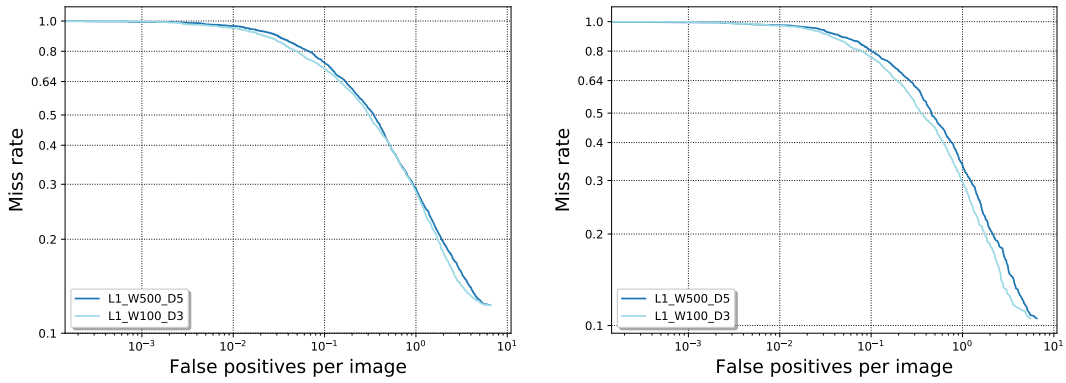
(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume, allPoints: keine Filterung durch Entropieschwellwert, max7m: im Training wurden nur Cluster bis zu einer Distanz von 7m verwendet



(a) (b)

Abbildung 5.7: DET-Kurve der AdaBoost-Klassifikatoren unter Verwendung des FPFH

(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume, max7m: im Training wurden nur Cluster bis zu einer Distanz von 7m verwendet



(a) (b)

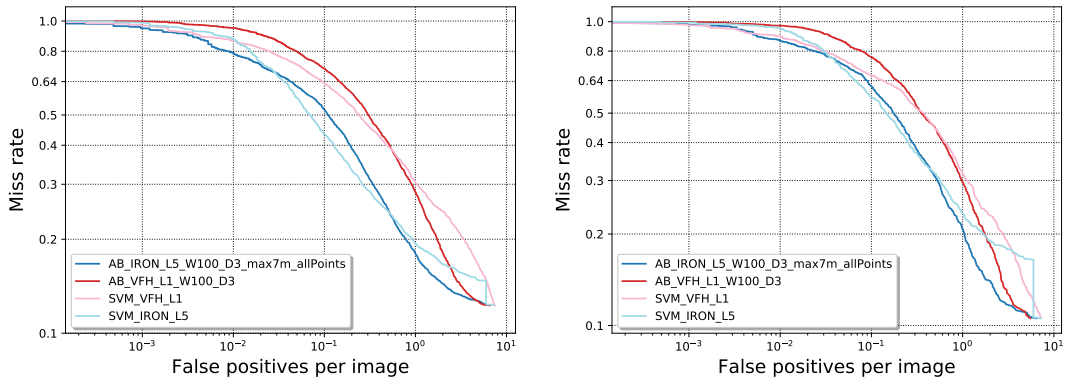
Abbildung 5.8: DET-Kurve der AdaBoost-Klassifikatoren unter Verwendung des FPFH

(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume

einer Schicht. Bei der verwendeten OpenCV2-Implementierung handelt es sich um keine parallelisierbare Berechnung des VFH. Somit wird bei mehreren Schichten die Berechnung sequentiell ausgeführt und bedeutet somit bei drei Schichten bereits eine Verdreifachung der Klassifikationszeit. Für eine Untersuchung der Einteilung in Schichten ist somit in zukünftigen Arbeiten eine erweiterte Implementierung erforderlich.
Die Miss Rate ist mit circa 70% sehr hoch. Da nur eine Schicht verwendet wird, ist anzunehmen, dass sich die VFH-Deskriptoren von Personen mit jeglichen Verdeckungen und Körperhaltungen nicht nur gering voneinander unterscheiden. Eine Separierung von den Negativbeispielen wird somit schwerer, als es vermutlich mit mehr als einer Schicht wäre.

5.5.2 Evaluation der SVM-Klassifikatoren

In ersten Tests wurde festgestellt, dass das Training der SVMs einen zu langen Zeitraum beansprucht, um zum Vergleich aller Features und Parameter verwendet zu werden. Die im Folgenden gezeigten Ergebnisse bestehen aus trainierten Klassifikatoren



(a) (b)

Abbildung 5.9: Vergleich von AdaBoost-Klassifikatoren mit SVMs
(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate, AB: AdaBoost, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume

zu einem Zeitpunkt, an dem der Trainingsdatensatz noch nicht vollständig war. Daher dienen die Schlussfolgerungen aus diesem Abschnitt dazu, abzuschätzen, ob es von Vorteil ist, zukünftig die besten evaluierten Featureparameter für das Training einer SVM zu nutzen.
Die Verwendung einer 2-Klassen-SVM mit IRON-Features bewirkt eine Verbesserung um 8,7% trotz einer geringeren Menge an Trainingsdaten. Bei der Verwendung des VFH als Deskriptor wurde eine Verbesserung von 7,1% durch die Verwendung einer SVM erreicht.
Eine SVM mit RBF-Kernel ist somit deutlich besser in der Lage, die Klassen mit IRON- oder VFH-Deskriptoren zu trennen als ein AdaBoost-Klassifikator (siehe Abbildung 5.9).

5.5.3 Zusammenfassung der Evaluation der eigenen Detektionssysteme

In den vorherigen Abschnitten wurden unterschiedliche Features auf ihre Eignung untersucht, Personen in diversen Körperhaltungen in einer Supermarktumgebung zu be-

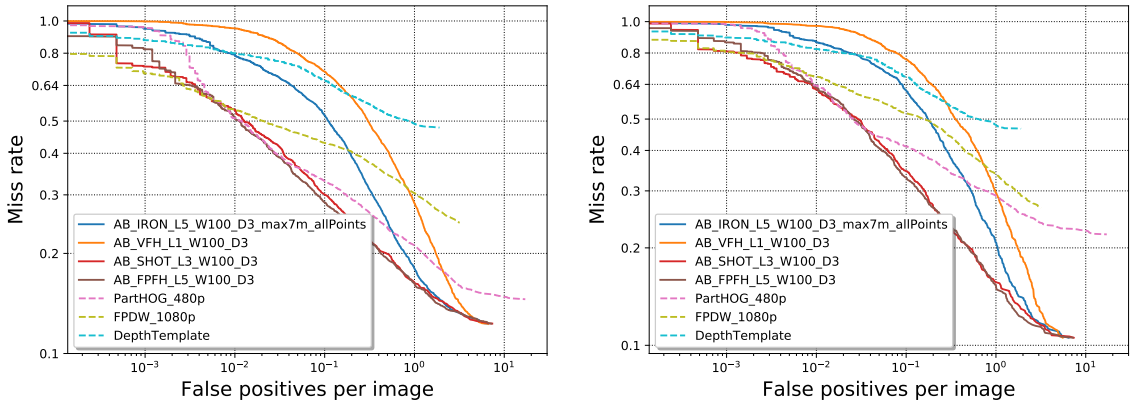
schreiben. An dieser Stelle erfolgt eine kurze Zusammenfassung der bisherigen Ergebnisse des eigenen Ansatzes.
Die Einteilung in mehrere Schichten hat sich durchgehend als ein positiver Einflussfaktor auf die Detektionsleistung ergeben. Je nach Feature ist eine andere Anzahl an Schichten sinnvoll, welches unter anderem in der Größe der jeweiligen Histogramme begründet sein kann. Das FPFH besteht aus 33 Bins, wodurch auch bei einer höheren Schichtenanzahl die Bins durch mehrere Punkte gefüllt werden. Daher verbessert sich vermutlich die DET-Kurve zwischen der Verwendung von drei und fünf Schichten im Unterschied zum SHOT-Feature (siehe Abb. 5.5), welcher aus insgesamt 352 Bins besteht.
Das Training von SVMs hat bei der Verwendung von IRON-Features zu einer deutliche Verbesserung geführt, daher sollte dieser Ansatz in zukünftigen Arbeiten weiter verfolgt werden. Eine Klassifikation der weiteren verwendeten Features durch eine SVM könnte somit eine Erhöhung der Detektionsleistung nach sich ziehen.
Des Weiteren wurde der Einfluss einer Begrenzung der Trainingsdaten auf eine maximale Distanz von sieben Metern untersucht. Lediglich bei den aktuell verwendeten Parametern für die IRON-Features wurde hiermit eine Verbesserung erreicht. Das Entfernen der Trainingsbeispiele führte bei der Klassifikation von anderen Features zu einer Verschlechterung der Detektionsleistung.
Eine Bewertung der Eignung des VFH kann auf Grundlage der nicht vorhandenen Evaluation für eine Einteilung in mehrere Schichten im Rahmen dieser Masterarbeit nicht erfolgen. So beträgt die Miss Rate bei der Verwendung des FPFH (keine Einteilung in Schichten) nur 1% weniger als bei der Nutzung des VFH-Deskriptors. Trotzdem ist bei einer Einteilung in Schichten der Klassifikator mit FPFH einer der Besten in dieser Evaluation.
Im folgenden Abschnitt werden die für die einzelnen Features jeweils besten Klassifikatoren mit den Referenzverfahren verglichen. Dafür wird aus jedem Diagramm der vorherigen Abschnitte die Parameterkombination gewählt, welche die geringste Miss Rate auf den *reasonable* Personen bis zu einer Distanz von 10 Metern bei 0.1 Falschdetektionen pro Bild besitzt.

5.5.4 Evaluation der Referenz-Detektionssysteme

Die drei Referenz-Detektionssystemen, die in Abschnitt 3.3 vorgestellt wurden, werden auf vier verschiedenen Auflösungsstufen angewandt. Das ursprüngliche RGB-Bild der Kinect2 besitzt eine Auflösung von 1290 × 1080 Pixeln (1080p). Um die Bewegungsunschärfe zu reduzieren, wird das Bild sowohl auf 720p (1280 × 720 Pixel) als auf 480p (854 × 480 Pixel) skaliert. Als vierte auszuwertende Auflösung wurde 360p (640 × 360 Pixel) gewählt, da dies am ehesten der Auflösung des Tiefenbildes der Kinect2 entspricht und somit eine äquivalente Auswertung auf Grundlage der Höhe einer 2D-Detektionsbox ermöglicht wird.
Die Auswahl der besten Auflösungsstufe erfolgt auf Grundlage der im Anhang B.3 dargestellten DET-Kurven.
Im Folgenden werden als Referenzverfahren der Tiefentemplate-Detektor (siehe Abschnitt 3.3.1), der FPDW in der Auflösungsstufe 1080p (siehe Abschnitt 3.3.4) und der PartHOG in der Auflösungsstufe 480p (siehe Abschnitt 3.3.3) verwendet.

5.5.5 Vergleich des SuPer-Detektors mit Referenzverfahren

Im Folgenden werden die jeweils besten Klassifikatoren aus den vorherigen Abschnitten mit den Referenzverfahren verglichen hinsichtlich der Klassifikationsleistung und Performanz.
Abbildung 5.10 beinhaltet die DET-Kurven für jedes untersuchte Feature. Dabei wurde zur Übersichtlichkeit jeweils die beste Parameterkombination der AdaBoost-Parameter und Feature-Parameter ausgewählt. Die Evaluationsergebnisse der drei Referenzverfahren sind jeweils in gestrichelten Linien dargestellt.
Die beste AdaBoost- und Featurekombination ergibt sich bei den *reasonable* Personen bis 10m bei der Verwendung des FPFH-Deskriptors in fünf Schichten mit einem AdaBoost-Klassifikator aus 100 Entscheidungsbäumen (Weak Learner) der maximalen Tiefe 3. Im Vergleich zu dem besten Referenzverfahren (PartHOG auf 480p) ergibt sich eine Verbesserung in der Miss Rate um 4,6%. Die Nutzung des SHOT-Deskriptors führt ebenso zu einer besseren Detektionsleistung als der PartHOG.
Somit besitzt der SuPer-Detektor eine bessere Klassifikationsleistung als alle verwen-



(a) (b)

Abbildung 5.10: Vergleich des SuPer-Detektors mit Referenzverfahren
(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate, AB: AdaBoost, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume, gestrichelte Kurve: Referenzverfahren, durchgezogene Kurve: eigene Ansätze

deten Referenzverfahren. Eine Erkennung von hockenden und stehenden Personen in einer Einkaufsmarktumgebung in 3D-Punktwolken bis zu einer Distanz von 10 Metern wurde somit erreicht.
Im folgenden Abschnitt wird der Super-Detektor zusätzlich hinsichtlich der möglichen Bildrate bewertet.

5.5.6 Auswertung der Klassifikationsdauer

In diesem Abschnitt wird die ermittelte Klassifikationsdauer des Detektionssystems mit den Referenzverfahren verglichen. Eine robuste Klassifikation ist ohne eine gute Geschwindigkeit zum Einsatz auf mobilen Robotern ungeeignet. Dies wird für die Navigation und Interaktion mit Nutzern benötigt, eine veraltete Detektion von Personen ist nicht hilfreich. Tabelle 5.3 zeigt die durchschnittlichen Klassifikationszeiten der zuvor evaluierten Detektoren pro Bild. Das schnellste Verfahren ist hierbei der Tiefentemplate-Detektor, welcher jedoch in der Klassifikationsleistung deutlich

	Durchschnitt	St.-Abw.
FPFH_L5_W100_D3	206	57
SHOT_L3_W100_D3	207	58
IRON_L5_W100_D3_max7m_allPoints	273	86
VFH_L1_W100_D3	157	51
FPDW_1080p	526	10
FPDW_720p	212	7
PartHOG_480p	338	19
PartHOG_360p	200	12
DepthTemplate	9	2

Tabelle 5.3: Zeitaufwand für die Klassifkation pro Bild
St.-Abw.: Standardabweichung

schlechter ist. Bei dem FPDW könnte die 720p-Auflösungsstufe in der Anwendungsphase gewählt werden, da die Klassifikationsleistung sehr ähnlich bleibt und lediglich 212ms benötigt. Der schnellste eigene Ansatz ist die Wahl des VFH als Deskriptor. Der beste eigene Klassifikator (siehe 5.10) verwendet das FPFH mit einer Einteilung in fünf Schichten. Die benötigte Zeit von durchschnittlich 206ms pro Bild führt zu einer möglichen Bildrate von circa 4 Bildern pro Sekunde. Ungefähr dieselbe Zeit benötigt der PartHOG in den geringeren Auflösungsstufen. Jedoch ist die Klassifikationsleistung des PartHOG geringer und die verwendete Implementierung lastet alle verfügbaren Kerne des Prozessors voll aus.
Somit besitzt der eigens entworfene SuPer-Detektor eine Klassifikationszeit pro Bild, mit welcher der Einsatz auf einem mobilen Roboter möglich ist. Im Folgenden wird die Klassifikationsdauer pro Bild für den SuPer-Detektor näher untersucht. Abbildung 5.11 zeigt die benötigte Rechenzeit der einzelnen Verarbeitungsschritte, welche in 4.1 dargestellt werden. Nahezu die Hälfte der Zeit wird für die Berechnung der Oberflächennormalen benötigt. 46% werden für die Berechnung des FPFH-Desktriptors verwendet. Dabei sind pro Bild im Durchschnitt 16 Kandidatencluster zu klassifizieren. Hieraus ergeben sich mehrere Ansätze, um die Gesamtdauer zu beschleunigen. In

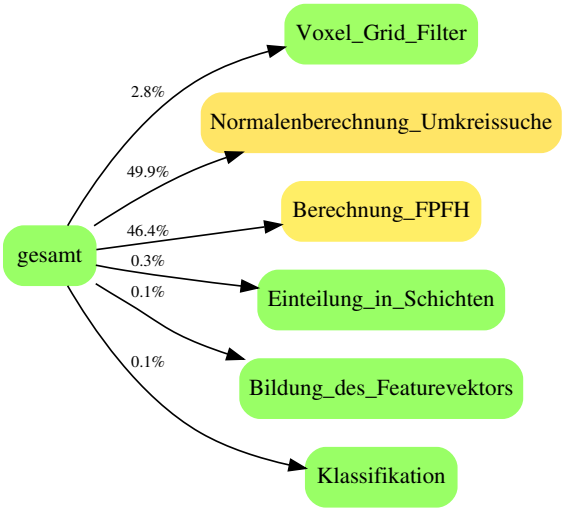


Abbildung 5.11: Untersuchung der Zeitdauer für einzelne Schritte des SuPer-Detektors (siehe 4.1)
Zeitmessung anhand von 500 Bildern des Testdatensatzes während der Fahrt des Roboters, verwendet wurde ein AdaBoost-Klassifikator mit 100 Weak Learnern der max. Tiefe 3 mit dem FPFH-Deskriptor in 5 Schichten

weiterführenden Experimenten wird im folgenden Abschnitt 5.6 untersucht, welche Alternativen bei der Bestimmung der Oberflächennormalen das Detektionsergebnis trotz schnellerer Berechnung nicht verschlechtern. Zudem wird evaluiert, 4.2.4 vorgestellte *No-Person-Map* geeignet ist, um die Anzahl der zu klassifizierenden Kandidatencluster zu verringern.

5.6 Weiterführende Experimente

Auf Grundlage der vorhergegangenen Evaluation wurden im Rahmen dieser Masterarbeit zusätzliche Experimente unternommen, um sowohl eine bessere Detektionsleistung als auch eine höhere Performanz des SuPer-Detektors zu erreichen. Die Ergebnisse der Untersuchungen werden im Folgenden erläutert und bewertet.

5.6.1 Distanzbasierte Auswertung

Die bisherige Auswertung erfolgte bis zu einer Distanz von 10 Metern, da dies das gegebene Ziel der Masterarbeit darstellt. Wie in Tabelle 5.1 ersichtlich, befinden sich ein Zehntel der zu detektierenden Personen in einer Entfernung zwischen 10 und 18 Metern. Nahezu 70% der Testdaten befindet sich in einer Entfernung bis zu 7 Metern. Daher wird für diese zwei Entfernungsbereiche eine weitere Evaluation durchgeführt, aus der ersichtlich wird, wie das eigens entwickelte Verfahren sich verhält.
Abbildung 5.12 zeigt die Ergebnisse der distanzbasierten Evaluation. Bei der Betrachtung der Klassifikationsleistung bis zu einer Distanz von 7 Metern ist der Part-HOG als Referenzverfahren deutlich näher an der Klassifikationsleistung des SuPer-Detektors als bei einer Distanz von bis zu 10m (siehe Abb. 5.10). Nichtsdestotrotz ist die Miss Rate des im Rahmen dieser Masterarbeit entwickelten Verfahrens um 1,1% geringer.
Bei Einbezug aller *reasonable* Personen bis 18 Metern kann ein großer Anteil durch das eigene Verfahren detektiert werden. Ebenso werden auch verdeckte Personen mit einer Miss Rate von 31,4% detektiert im Vergleich zum Part-HOG mit 38,3%.
Mit dem in dieser Masterarbeit entworfenen SuPer-Detektor ist eine Erkennung von Personen in einer Einkaufsmarktumgebung somit über eine Distanz von 10 Metern hinaus möglich.

5.6.2 Stehend oder Hockend?

Im Projekt ROTATOR (siehe Abschnitt 1.2) muss der Roboter entscheiden, in welcher Art und Weise mit Personen interagiert, die ihn an daran hindern, Regale zu scannen. Wenn eine Person mit einem Einkaufswagen durch den Gang geht oder vor einem Regal steht, könnte eine höfliche Bitte bereits ausreichen, damit der Weg frei gemacht wird. Hockt eine Person jedoch vor einem Regal, ist anzunehmen, dass diese aktuell beschäftigt ist und sich zudem nicht ohne Weiteres von der Stelle bewegen kann. Hier wäre zum Beispiel die Entscheidung möglich, die Person nicht anzusprechen und zu einem späteren Zeitpunkt zu der entsprechenden Regalreihe zurückzukehren. Somit wird deutlich, dass über eine reine Personendetektion es hilfreich ist, mehr Informationen über die Körperhaltung einer Person zu erhalten.

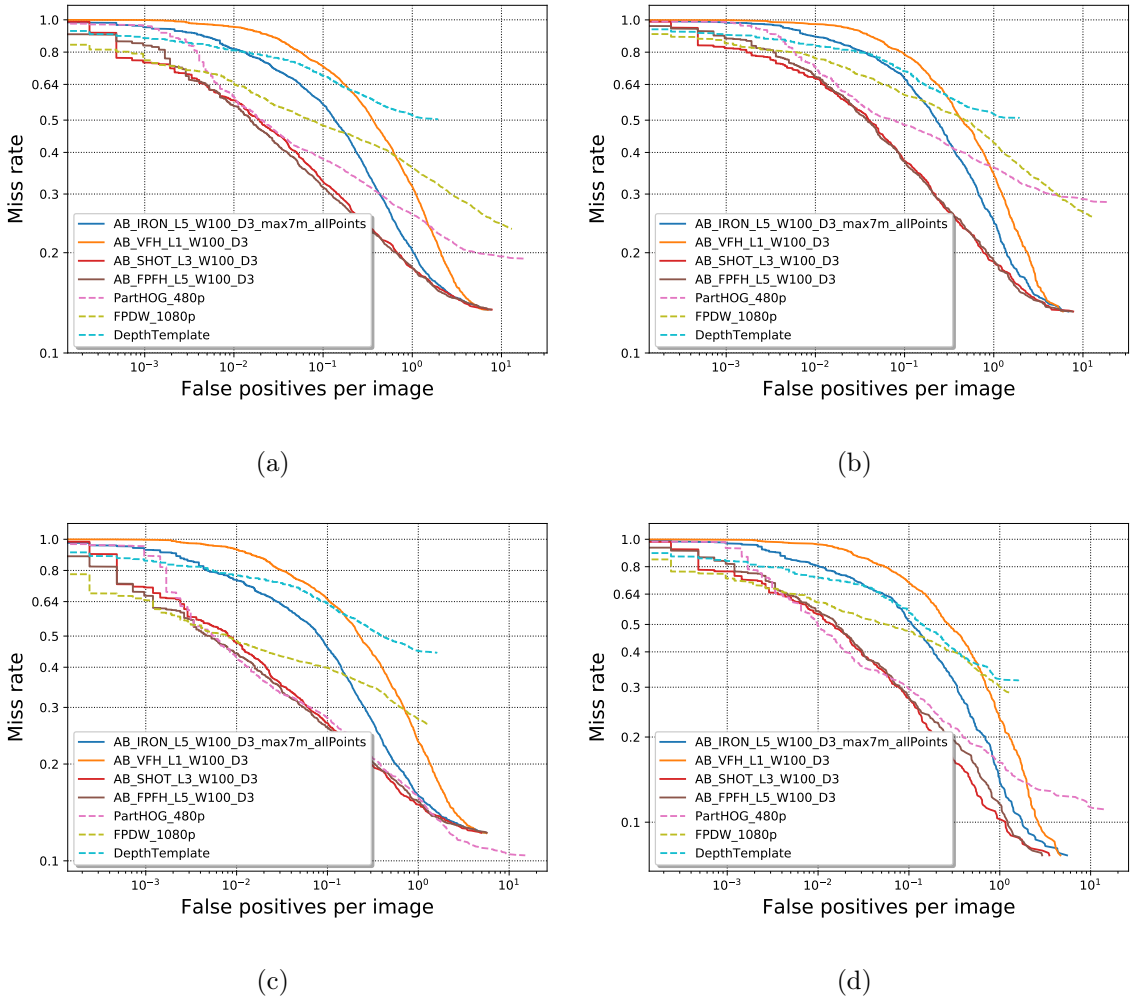


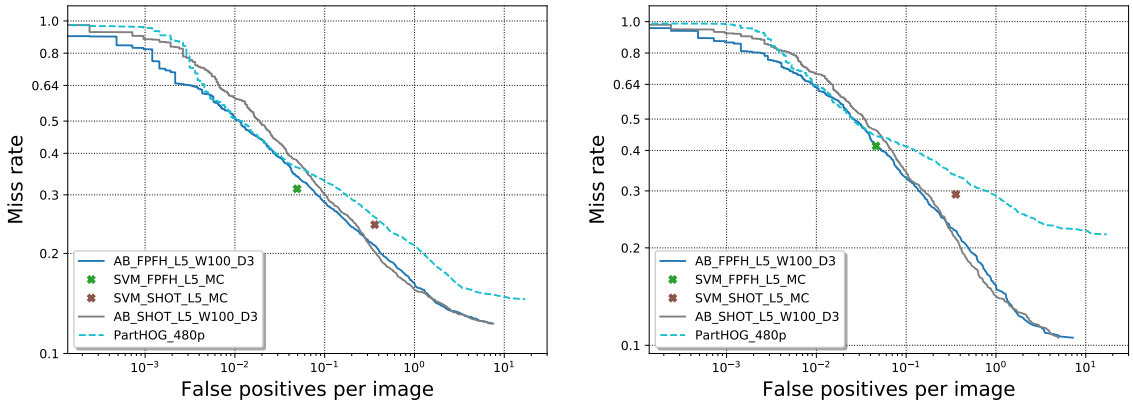
Abbildung 5.12: Vergleich des SuPer-Detektors mit Referenzverfahren

(a) reasonable bis 18m, (b) occluded bis 18m, (c) reasonable bis 7m, (d) occluded bis 7m, x-Achse: False Positives Per Image, y-Achse: Miss Rate, AB: AdaBoost, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume

(a) (b)

Abbildung 5.13: Klassifikation von stehenden und hockenden Personen
(a) Farbbild einer szenariotypischen Situation, (b) grün: detektierte stehende Person, gelb: hockende detektierte Person

Daher wurde im Rahmen dieser Masterarbeit untersucht, ob eine Unterscheidung von hockenden und stehenden Personen durch einen Klassifikator möglich ist. Die trainierten Modelle stammen wie die 2-Klassen-SVMs aus einem kleineren Trainingsdatensatz (siehe Abschnitt 5.5.2). Als Features wurden hierfür zunächst das FPFH und der SHOT-Deskriptor mit einer Einteilung in fünf Schichten verwendet. Als Klassifikator wird eine Multi-Klassen-SVM (siehe Abschnitt 3.6) trainiert zur Unterscheidung der Klassen *stehende Person*, *nicht stehende Person* und *keine Person*. Da die Implementierung der Multi-Klassen-SVM als Ergebnis keine Konfidenz besitzt, kann keine komplette Kurve in das DET-Diagramm eingezeichnet werden. Abbildung 5.13 zeigt eine stehende und eine hockende Person, welche durch die Klassifikation einer Multi-Klassen-SVM korrekt entsprechend ihrer Pose klassifiziert werden.
In Abbildung 5.14 wird deutlich, dass die Klassifikationsleistung durch Multi-Klassen-SVMs vergleichbar ist mit der durch AdaBoost-Klassifikatoren. In fortführenden Arbeiten kann untersucht werden, wie präzise die Multi-Klassen-SVM *stehende* und *nicht stehende* Personen trennt.



(a) (b)

Abbildung 5.14: Vergleich der Klassifikation durch AdaBoost-Klassifikatoren, PartHOG und Multi-Klassen-SVMs

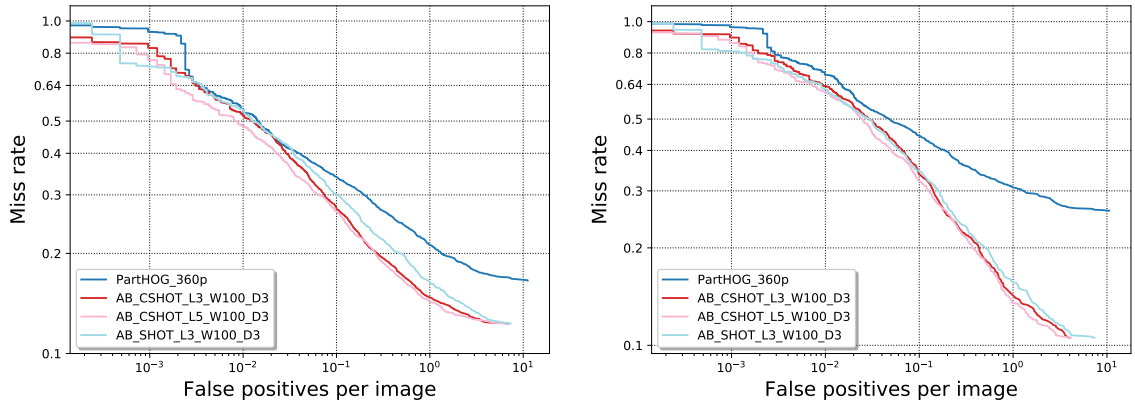
(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate, AB: AdaBoost, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume

5.6.3 Einbezug von Farbe

Bildbasierte Referenzverfahren verwenden Farbinformationen, um Personen von anderen Objekten zu unterscheiden. Sowohl der PartHOG-Detektor als auch der FPDW sind in der Lage, Personen in dem vorliegenden Szenario Supermarkt zu detektieren. Daher wurde im Rahmen dieser Masterarbeit untersucht, ob eine Kombination des SuPer-Detektors mit weiteren Features auf Farbe basierend einen Vorteil für die Detektion birgt. Zur Bewertung wurde als Feature der Color-SHOT (siehe Abschnitt 3.1.1) verwendet mit AdaBoost als Klassifikator. In 5.15 wird deutlich, dass der Einbezug von Farbe die Klassifikationsleistung weiter verbessert, im Vergleich zum bisherigen Klassifikator (mit Verwendung des reinen SHOT-Deskriptors) um weitere 3%.

5.6.4 Normalenberechnung durch kNN

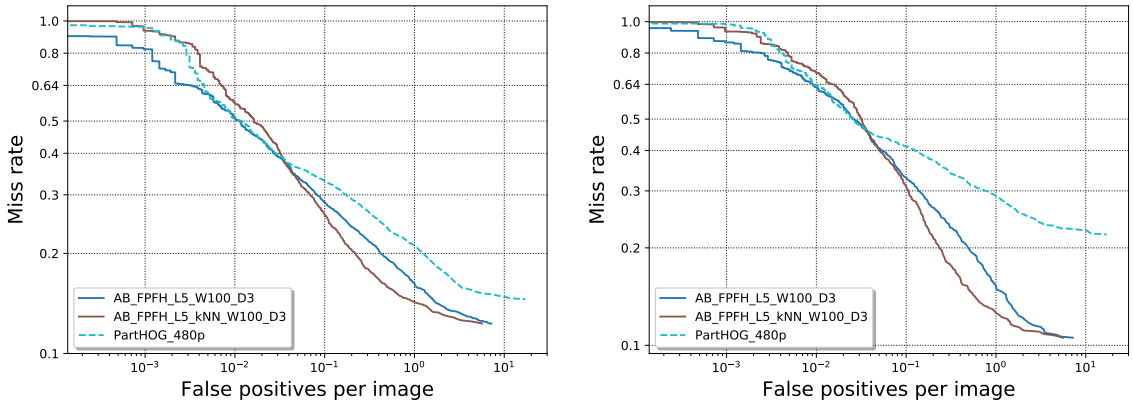
Der bisher beste SuPer-Detektor verwendet das FPFH als Deskriptor bei einer Einteilung in fünf Schichten (siehe Abb. 5.10). Die für die Berechnung der Oberflächennor-



(a) (b)

Abbildung 5.15: Vergleich der DET-Kurven mit Einbezug von Farbe
(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate, AB: AdaBoost, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume

malen notwendigen benachbarten Punkte wurden in den bisherigen Untersuchungen über eine Umkreissuche bestimmt. Dies kann wie in Abschnitt 5.3.1 beschrieben auch durch die Suche nach den k nächsten Nachbarn erreicht werden. In Abb. 5.16 wird deutlich, dass bei einer Nachbarschaftsbestimmung (hier k=25) eine Verbesserung um 2,3% erreicht wird. Dies wird eventuell durch einen zu großen Radius bei der Umkreissuche verursacht, welches in weiteren Experimenten bestätigt werden kann. Zudem ist in zukünftigen Arbeiten der Einfluss des k näher zu untersuchen und eine optimale Anzahl an zu bestimmenden Nachbarn herauszufinden.
Zusätzlich zur besseren Klassifikationsleistung ergibt ein Geschwindigkeitsvorteil von durchschnittlich 39 Millisekunden pro Bild im Vergleich zur Nachbarschaftsbestimmung durch eine Umkreissuche. Somit kann nun eine Geschwindigkeit von 167ms (mit einer Standardabweichung von 43ms) und erreicht werden, woraus sich eine Bildrate von 5 Bildern pro Sekunde ergibt.

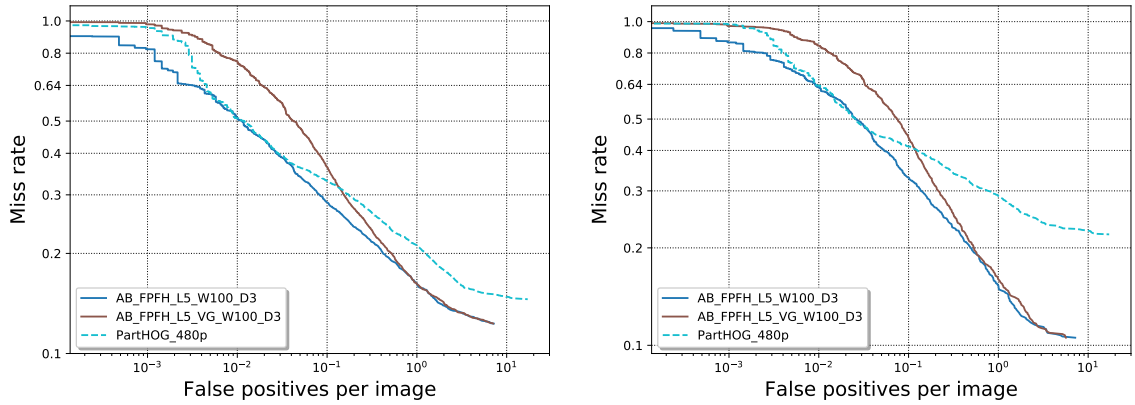


(a) (b)

Abbildung 5.16: Vergleich der Nachbarschaftsbestimmung bei der Normalenberechnung durch Umkreissuche und *kNN*
(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate, AB: AdaBoost, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume, kNN: Berechnung der Nachbarschaft durch kNN

5.6.5 Normalenberechnung auf gefilterten Clustern

Wie in Abschnitt 5.10 dargestellt, wurden bei der Normalenberechnung in den vorherigen Abschnitten jeweils die originalen Cluster verwendet. Eine weitere Möglichkeit besteht in der Berechnung der Oberflächennormalen auf den durch ein Voxel Grid (siehe Abschnitt 4.2.5) gefilterten Clustern. Hierfür wird jedes Cluster zunächst gefiltert und die Nachbarschaftsbeziehungen innerhalb des gefilterten Clusters ausgewertet.
Abbildung 5.17 zeigt die Ergebnisse, aus denen ersichtlich ist, dass die Detektionsleistung um 7,9% abnimmt. Da die Voxel mit einer Größe von $6 \times 6 \times 6$ Zentimetern gewählt wurden, fließen weniger Punkte bei der Umkreissuche in die Normalenberechnung mit ein. Hiermit gehen Informationen verloren, die bei gleichbleibendem Radius der Umkreissuche von Bedeutung sein können.
Die Klassifikationsgeschwindigkeit nimmt in diesem Ansatz um 60 Millisekunden auf 146ms pro Bild ab, da deutlich weniger Punkte bei der Nachbarschaftssuche betrachtet werden.



(a) (b)

Abbildung 5.17: Vergleich der Nachbarschaftsbestimmung bei der Normalenberechnung durch Umkreissuche und *kNN*
(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate, AB: AdaBoost, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume, VG: Berechnung der Nachbarschaft auf gefilterten Clustern

5.6.6 Nutzung einer *No-Person-Map*

Die Verarbeitungszeit pro Bild ist beim SuPer-Detektor stark abhängig von der Anzahl der zu klassifizierenden Kandidatencluster. Je weniger Cluster klassifiziert werden müssen, desto schneller wird das Detektionssystem. In Abschnitt 4.7 wurde der Ansatz entwickelt, Kandidatencluster direkt in als *keine Person* zu klassifizieren, wenn ein Großteil in einem Bereich liegt, in welchem keine Personen vorhanden sind. Dieser Ansatz wurde während der Erstellung des Testdatensatzes entwickelt und implementiert. Daher kann dieser Ansatz nicht auf dem kompletten Testdatensatz evaluiert werden, da hierfür in Teilen wichtige Informationen nicht vorhanden sind.
Daher findet die Evaluation in Bezug auf die benötigte Zeitdauer für die Klassifikation eines Bildes statt. Die durchschnittliche Klassifikationszeit für alle Cluster eines Bildes beträgt im Schnitt 127 Millisekunden. Im Vergleich dazu wird ohne Verwendung der *No-Person-Map* im Schnitt eine Zeit von 196 Millisekunden benötigt. Die zugrunde liegende *No-Person-Map* ist in Abb. 4.7(b) dargestellt.

Eine Verwendung der Lokalisation des Roboters in Kombination mit einer *No-Person-Map* erwirkt somit einen Vorteil bei der Geschwindigkeit des SuPer-Detektors.

5.7 Zusammenfassung

In diesem Kapitel wurden die im Rahmen dieser Masterarbeit entworfenen Ansätze zur Detektion von stehenden und hockenden Personen in einem Supermarkt evaluiert. Dabei wurden zunächst die verschiedenen Features auf ihre Eignung für das gegebene Einsatzszenario untersucht und im Anschluss mit den Referenzverfahren verglichen. Die Einteilung in Schichten führte dabei zu einer deutlichen Verbesserung der Klassifikationsleistung. Der FPFH-Deskriptor sowie der SHOT-Deskriptor mit Einbezug von Farbe erwiesen sich im Rahmen dieser Masterarbeit als am geeignetsten zur Personendetektion.
Weiterführende Experimente haben zu einer weiteren Verbesserung der Klassifikation und der Zeitdauer für die Verarbeitung eines Bildes geführt. Im Anhang B.5 sind zusätzlich die Auswertungen der Verfahren auf dem gesamten SuPer-Datensatz ersichtlich.

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

In dieser Masterarbeit wurde ein neues Verfahren zur Detektion von stehenden und hockenden Personen in einer Einkaufsmarktumgebung unter Verwendung von 3D-Punktwolken vorgestellt. Mit dem SuPer-Detektor konnten bessere Detektionsergebnisse als mit State of The Art Ansätzen erreicht werden. Es wurde der SuPer-Datensatz erstellt, ein Testdatensatz bestehend aus 4161 Bildern mit szenariotypischen Körperhaltungen von Personen. Eine Erkennung von Personen ist in 3D-Punktwolken nicht nur bis zu einer Entfernung von 10 Metern möglich, sondern in einem gewissen Rahmen auch darüber hinaus.
In weiterführenden Experimenten konnte nicht nur eine Person detektiert werden, sondern auch eine Unterscheidung bezüglich ihrer Körperhaltung getroffen werden. Somit ist es möglich, zu erkennen, ob eine Person steht oder hockt. Dies bedeutet einen erheblichen Vorteil für die sozialverträgliche, nutzerorientierte Navigation eines mobilen Roboters.
Eine Analyse der durchschnittlich benötigten Zeit für die Verarbeitung eines Bildes ergab ohne Nutzung einer *No-Person-Map* eine Bildrate von 5 Bildern pro Sekunde. Somit kann der SuPer-Detektor im Rahmen des Projektes ROTATOR auf einem mobilen Roboter eingesetzt werden.

6.2 Ausblick

Im Rahmen dieser Masterarbeit wurde aufgezeigt, dass eine Detektion von stehenden und hockenden Personen in einer Einkaufsmarktumgebung unter Verwendung von 3D-Punktwolken möglich ist. Es wurden weitere Ansätze gesammelt, die zu einer Verbesserung bzw. Performanzerhöhung des Klassifikators führen könnten. Diese werden im Folgenden kurz dargestellt und können als Grundlage für weiterführende Arbeiten dienen.

Aufnahme weiterer Trainingsdaten

Der Trainingsdatensatz beinhaltet aktuell nur einen geringen Anteil an nicht stehenden Personen aus einem kleinen Personenkreis. Um das breite Spektrum der möglichen Körperhaltungen abzudecken und eine bessere Generalisierungsfähigkeit zu erreichen, könnte die Aufnahme von weiteren Beispielen nicht stehender Personen in den Trainingsdatensatz vorteilhaft sein. Ebenso kann in weiteren Arbeiten untersucht werden, ob der Einbezug von Trainingsbeispielen in Entfernungen als mehr 12 Metern die Klassifikation von entfernten Personen verbessert.

Verwendung des Klassifikators als Verifikation

Die Nutzung des Blob-Extractors zur Segmentierung von Kandidatenclustern führt im gegebenen Szenario nicht zu einer perfekten Trennung der Punktwolke in Cluster, die nur aus Negativpunkten oder nur zu einer Person gehörend bestehen. Dieses könnte durch ein anderes Verfahren ersetzt werden. Der Tiefentemplate-Detektor von [Jafari et al., 2014] ist ein sehr schneller Personendetektor mit einer hohen Anzahl an Falsch-Positiv-Detektionen. Der im Rahmen dieser Masterarbeit entwickelte Klassifikator könnte an dieser Stelle die Detektionen verifizieren und somit eine robustere Detektionsleistung erzielen.

Einbezug von Farbe

In Abschnitt 5.15 ist ersichtlich, dass der Einbezug von Farbe die Detektionsleistung verbessert. In zukünftigen Untersuchungen kann diese Information verwendet werden,

um andere Deskriptoren mit Farbfeatures zu verknüpfen und somit eine weitere Verbesserung des Detektionssystems zu erreichen.

Klassifikation von unterschiedlichen Körperhaltungen

Im Rahmen dieser Masterarbeit wurde aufgezeigt, dass eine Klassifikation in *stehende* und *nicht stehende* Person möglich ist. Eine weitere Untersuchung, ob eine Einteilung in mehr Klassen möglich ist, erscheint somit sinnvoll. So könnte eine Multi-Klassen-SVM zur Trennung von mehr als zwei Körperhaltungen trainiert werden und somit eine Unterscheidung zwischen einer *stehenden*, *hockenden* und *anderen* Körperhaltung erreicht werden. Die Beschreibung der einzelnen Klassen ist in Abbildung 4.2.1 dargestellt.

Weitere Untersuchung der Feature-Parameter

In den weiterführenden Experimenten wurde festgestellt, dass die Verwendung des *kNN-Verfahrens* zur Bestimmung der Nachbarschaft (mit $k = 25$) sowohl eine Beschleunigung als auch Verbesserung der Detektionsleistung zur Folge hat. In fortführenden Experimenten kann nun eine Evaluation der Variation des Parameters k untersucht werden. Ebenso kann auf Grundlage dieser Masterarbeit eine experimentelle Bestimmung der optimalen Parameter für die Segmentierung und Parameter der weiteren verwendeten Features durchgeführt werden.

Verfeinerung der Umgebungskarte

Abbildung 4.7(b) zeigt die für die Anwendungsphase entworfene *No-Person-Map*. Zu ersten Tests wurden die festen Regalstrukturen nur in einer groben Abschätzung markiert. Eine genauere Markierung würde deutlich mehr Kandidatencluster entfernen und somit die Bildrate weiter erhöhen. Aktuell wird eine Überlappung der auf die *No-Person-Map* projizierten Bounding Box mit den markierten Regionen von 50% verwendet, eine experimentelle Bestimmung dieses Parameters kann ebenso in weiterführenden Arbeiten betrachtet werden.

Anhang A

Aufteilung des SuPer-Datensatzes

	0m - 7m	**7m - 10m**	**10m - 18m**	**gesamt**
stehend	287	182	175	644
hockend	0	0	0	0
andere	51	76	55	182
gesamt	338	258	230	826

Tabelle A.1: Anzahl Personen im SuPer-Datensatz während der Fahrt

	0m - 7m	**7m - 10m**	**10m - 18m**	**gesamt**
stehend	1736	396	198	2330
hockend	580	127	0	707
andere	323	110	7	440
gesamt	2639	633	205	3477

Tabelle A.2: Anzahl Personen im SuPer-Datensatz bei stehendem Roboter

Anhang B

Evaluation

B.1 Übersicht über trainierte Klassifikatoren

In den hier dargestellten Tabellen wird ein Überblick über die trainierten Klassifikatoren unter Verwendung des FPFH (Tabelle B.1), des IRON-Deskriptors (Tabelle B.2), des SHOT-Deskriptor (Tabelle B.3), sowie des VFH (Tabelle B.4) gegeben.

Schichten	**WL**	**max. Baumtiefe**	**max. Entfernung**
1	100	3	18.0
3	100	3	18.0
5	100	3	18.0
5	500	5	18.0
5	100	3	7.0
7	100	3	18.0

Tabelle B.1: Trainierte AdaBoost-Klassifikatoren mit FPFH
WL: Anzahl Weak Learner, max. Entfernung in Metern

Schichten	WL	max. Baumtiefe	max. Entfernung	min. Entropie
1	100	3	18.0	0.9
3	100	3	18.0	0.9
5	100	3	18.0	0.9
1	100	3	18.0	0.0
3	100	3	18.0	0.0
5	100	3	18.0	0.0
5	100	3	7.0	0.0
1	500	5	18.0	0.9
3	500	5	18.0	0.9
5	500	5	18.0	0.9
1	500	5	18.0	0.7
3	500	5	18.0	0.7
5	500	5	18.0	0.7

Tabelle B.2: Trainierte AdaBoost-Klassifikatoren mit IRON-Features
WL: Anzahl Weak Learner, max. Entfernung in Metern

Schichten	WL	max. Baumtiefe	max. Entfernung
1	100	3	18.0
3	100	3	18.0
3	100	3	7.0
5	100	3	18.0
1	500	5	18.0
3	500	5	18.0
5	500	5	18.0

Tabelle B.3: Trainierte AdaBoost-Klassifikatoren mit SHOT-Deskriptoren
WL: Anzahl Weak Learner, max. Entfernung in Metern

Schichten	WL	max. Baumtiefe	max. Entfernung
1	100	3	18.0
3	100	3	18.0
1	500	5	18.0

Tabelle B.4: Trainierte AdaBoost-Klassifikatoren mit VFH
WL: Anzahl Weak Learner, max. Entfernung in Metern

B.2 Evaluation der Klassifikationsdauer

	Durchschnitt	Standardabweichung
L1_W100_D3	204	57
L3_W100_D3	206	57
L5_W100_D3	206	57
L7_W100_D3	196	57

Tabelle B.5: Zeitaufwand für die Klassifkation pro Bild (in ms) bei Verwendung des FPFH-Deskriptors
L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume

	Durchschnitt	Standardabweichung
L1_W100_D3	208	58
L1_W500_D5	281	90
L3_W100_D3	207	58
L3_W500_D5	260	104
L5_W100_D3	213	59
L5_W500_D5	282	111

Tabelle B.6: Zeitaufwand für die Klassifkation pro Bild (in ms) bei Verwendung des SHOT-Deskriptors
L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume

B.3 Evaluation der Referenz-Detektionssysteme

B.3.1 Tiefentemplate

Die mittlere Berechnungsdauer des Tiefentemplate-Detektors beträgt pro Bild 9ms mit einer Standardabweichung von 2ms.

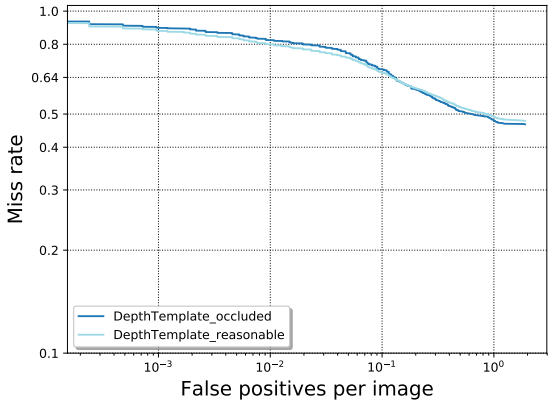
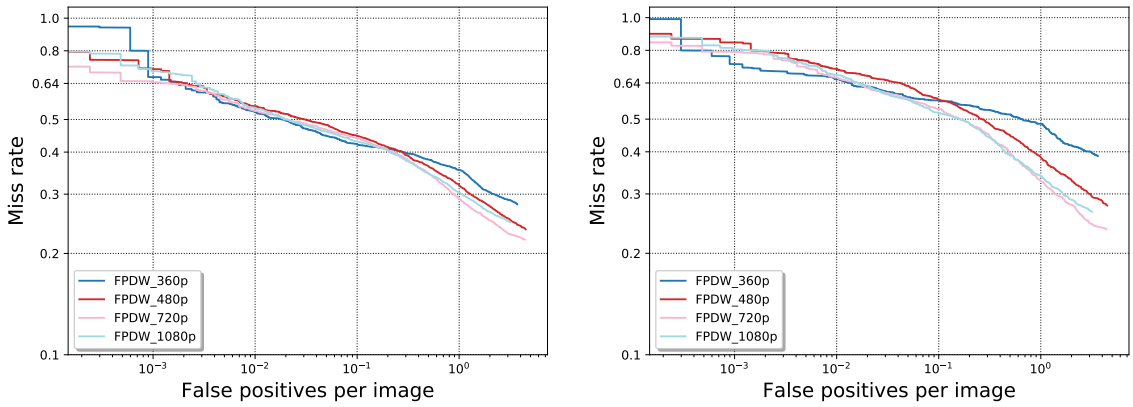


Abbildung B.1: DET-Kurve verschiedener Auflösungsstufen unter Verwendung des Tiefentemplate-Detektors
x-Achse: False Positives Per Image, y-Achse: Miss Rate

B.3.2 FPDW

In der Evaluationsklasse *reasonable* ist der FPDW mit einer Bildauflösung von 360p am besten, bei der Detektion von verdeckten Personen jedoch eignet sich die Verwendung der hochaufgelösten 1080p Bilder besser. Vergleicht man die Ergebnisse auf dem kompletten Datensatz, ergibt sich eine bessere Klassifikationsleistung in der 1080p-Auflösungsstufe. Diese wird somit in den Vergleichen mit den eigenen Ansätzen verwendet.



(a) (b)

Abbildung B.2: DET-Kurven verschiedener Auflösungsstufen unter Verwendung des FPDW

(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate

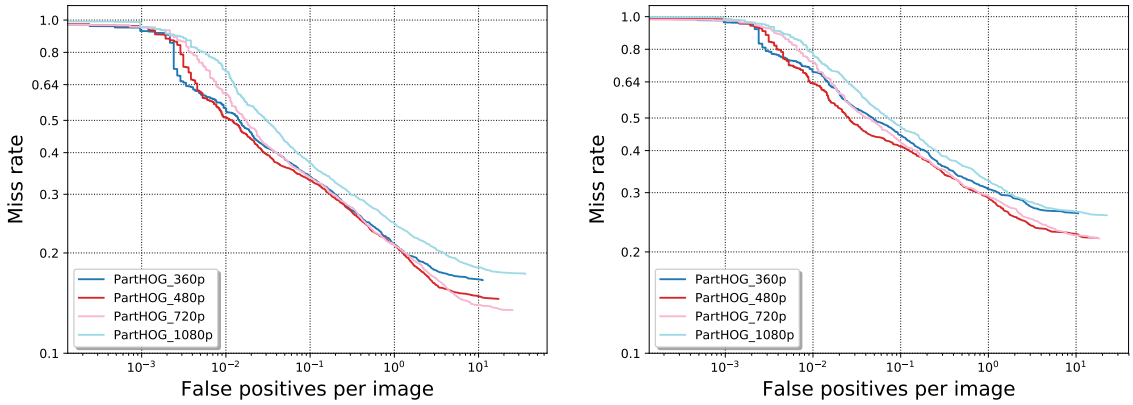
	Durchschnitt	**Standardabweichung**
360p	79	12
480p	90	3
720p	212	7
1080p	526	10

Tabelle B.7: Zeitaufwand für die Klassifkation pro Bild unter Verwendung des FPDW

Zeiten in Millisekunden

B.3.3 PartHOG

Sowohl für die Detektion der verdeckten Personen als auch der *reasonable* Evaluationsklasse eignet sich die Auflösungsstufe 480p. Jedoch ist die geringere Auflössungsstufe 360p nur minmal schlechter und bietet einen Geschwindigkeitsvorteil von 138ms pro Bild.



(a) (b)

Abbildung B.3: DET-Kurve verschiedener Auflösungsstufen unter Verwendung des PartHOG

(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate

	Durchschnitt	**Standardabweichung**
360p	200	12
480p	338	19
720p	707	27
1080p	1667	83

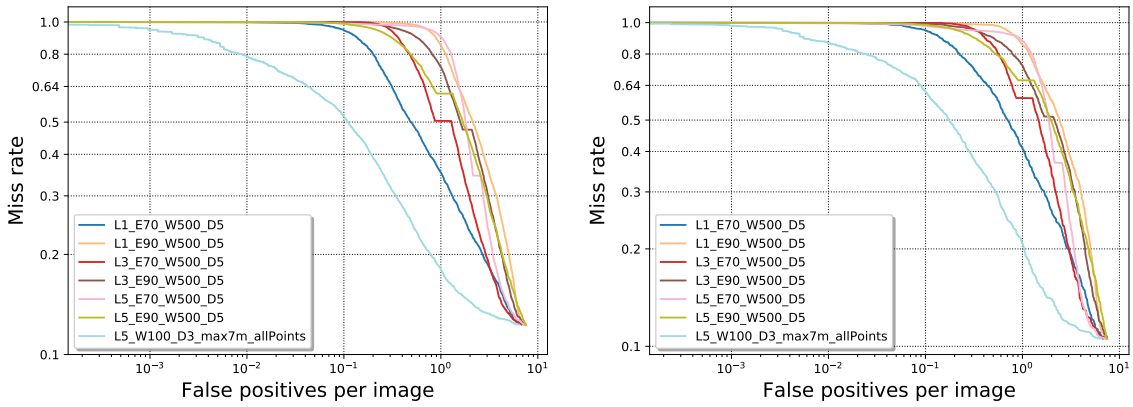
Tabelle B.8: Zeitaufwand für die Klassifkation pro Bild unter Verwendung des PartHOG

Zeiten in Millisekunden

B.4 Weitere Auswertungen

B.4.1 Evaluation des IRON-Deskriptors

Tabelle B.4 zeigt die Auswertung der AdaBoost-Klassifikatoren mit 500 Weak Learnern bei Verwendung des IRON-Deskriptors.



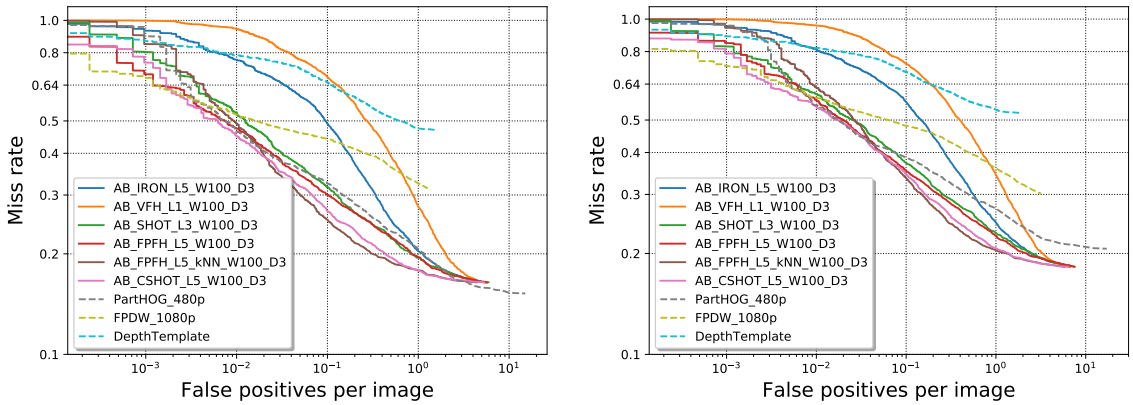
(a) (b)

Abbildung B.4: DET-Kurve verschiedener Auflösungsstufen unter Verwendung des IRON

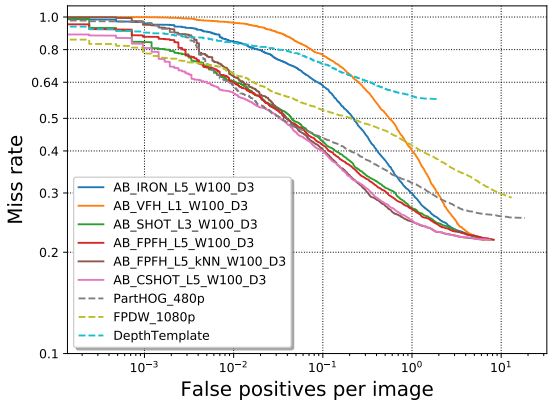
(a) reasonable, (b) occluded, x-Achse: False Positives Per Image, y-Achse: Miss Rate, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume

B.4.2 Distanzbasierte Auswertung

In diesem Abschnitt sind die Auswertungskurven der eigenen Ansätze und Referenzverfahren auf dem gesamten SuPer-Datensatz ersichtlich. Es werden somit alle Ground-Truth-Daten verwendet unabhängig von der Größe der Detektionsbox und des Verdeckungsgrades.



(a) (b)



(c)

Abbildung B.5: Vergleich des SuPer-Detektors mit Referenzverfahren (overall) *(a) overall bis 7m, (b) overall bis 10m, (c) overall bis 18m, x-Achse: False Positives Per Image, y-Achse: Miss Rate, AB: AdaBoost, L: Anzahl Schichten, W: Anzahl Weak Learner, D: max. Tiefe der Entscheidungsbäume, gestrichelte Kurve: Referenzverfahren, durchgezogene Kurve: eigene Ansätze, IRON: max. Trainingsdistanz 7m und keine Entropiefilterung*

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[Boser et al., 1992] Boser, Bernhard E., I. M. Guyon und V. N. Vapnik (1992). *A Training Algorithm for Optimal Margin Classifiers*. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, S. 144–152, New York, NY, USA. ACM.

[Carballo et al., 2014] Carballo, A., A. Ohya und S. Yuta (2014). *Fusion of double layered multiple laser range finders for people detection from a mobile robot*. In: *Multisensor Fusion and Integration for Intelligent Systems*, S. 5636–5643.

[Cho et al., 2012] Cho, H., P. E. Rybski, A. Bar-Hillel und W. Zhang (2012). *Real-time pedestrian detection with deformable part models*. In: *2012 IEEE Intelligent Vehicles Symposium*, S. 1035–1042.

[Choi et al., 2013] Choi, B., Ç. Meriçli, J. Biswas und M. Veloso (2013). *Fast human detection for indoor mobile robots using depth images*. In: *2013 IEEE International Conference on Robotics and Automation*, S. 1108–1113.

[Coates und Ng, 2011] Coates, Adam und A. Ng (2011). *The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization*. In: Getoor, Lise und T. Scheffer, Hrsg.: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, S. 921–928, New York, NY, USA. ACM.

[Dalal und Triggs, 2005] Dalal, N. und B. Triggs (2005). *Histograms of oriented gradients for human detection*. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Bd. 1, S. 886–893 vol. 1.

[Dasarathy, 1991] Dasarathy, Belur V (1991). *Nearest Neighbor (NN) Norms NN pattern Classification Techniques*.

[Dollar et al., 2012] Dollar, P., C. Wojek, B. Schiele und P. Perona (2012). *Pedestrian Detection: An Evaluation of the State of the Art*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(4):743–761.

[Dollar et al., 2010] Dollar, Piotr, S. Belongie und P. Perona (2010). *The Fastest Pedestrian Detector in the West*. In: *Proceedings of the British Machine Vision Conference*, S. 68.1–68.11. BMVA Press. doi:10.5244/C.24.68.

[Dollar et al., 2009] Dollar, Piotr, Z. Tu, P. Perona und S. Belongie (2009). *Integral Channel Features*. In: *Proceedings of the British Machine Vision Conference*, S. 91.1–91.11. BMVA Press. doi:10.5244/C.23.91.

[Dubout und Fleuret, 2012] Dubout, Charles und F. Fleuret (2012). *Exact Acceleration of Linear Object Detectors*. In: *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12, S. 301–311, Berlin, Heidelberg. Springer-Verlag.

[Ester et al., 1996] Ester, Martin, H.-P. Kriegel, J. Sander und X. Xu (1996). *A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, S. 226–231. AAAI Press.

[Felzenszwalb et al., 2010] Felzenszwalb, P. F., R. B. Girshick, D. McAllester und D. Ramanan (2010). *Object Detection with Discriminatively Trained Part-Based Models*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645.

[Felzenszwalb und Huttenlocher, 2004] Felzenszwalb, Pedro F. und D. P. Huttenlocher (2004). *Efficient Graph-Based Image Segmentation*. Int. J. Comput. Vision, 59(2):167–181.

[Freund und Schapire, 1997] Freund, Yoav und R. Schapire (1997). *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. Journal of computer and system sciences, 55:119–139.

[Garcia-Garcia et al., 2016] Garcia-Garcia, A., F. Gomez-Donoso, J. Garcia-Rodriguez, S. Orts-Escolano, M. Cazorla und J. Azorin-Lopez (2016). *PointNet: A 3D Convolutional Neural Network for real-time object class recognition*. In: *2016 International Joint Conference on Neural Networks (IJCNN)*, S. 1578–1584.

[Gruen et al., 2002] Gruen, Thomas W., D. Corsten und S. Bharadwaj (2002). *Retail Out of Stocks: A Worldwide Examination of Causes, Rates, and Consumer Responses*. Grocery Manufacturers of America.

[Hegger et al., 2013] Hegger, Frederik, N. Hochgeschwender, G. K. Kraetzschmar und P. G. Ploeger (2013). *People Detection in 3d Point Clouds Using Local Surface Normals*, S. 154–165. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Jafari et al., 2014] Jafari, O. H., D. Mitzel und B. Leibe (2014). *Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras*. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, S. 5636–5643.

[Kidono et al., 2011] Kidono, Kiyosumi, T. Miyasaka, A. Watanabe, T. Naito und J. Miura (2011). *Pedestrian recognition using high-definition LIDAR*. In: *IEEE Intelligent Vehicles Symposium*, S. 405–410.

[Krizhevsky et al., 2012] Krizhevsky, Alex, I. Sutskever und G. E. Hinton (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, S. 1097–1105, USA. Curran Associates Inc.

[Lawin et al., 2016] Lawin, Felix Järemo, P.-E. Forssén und H. Ovrén (2016). *Efficient Multi-Frequency Phase Unwrapping using Kernel Density Estimation*. In:

European Conference on Computer Vision (ECCV), Amsterdam. Springer International Publishing AG. VR Projects: Learnable Camera Motion Models, 2014-5928, Energy Models for Computational Cameras, 2014-6227.

[Lewandowski et al., 2017] Lewandowski, B., T. Wengefeld, T. Schmiedel und H. M. Gross (2017). *I see you lying on the ground - Can I help you? Fast fallen person detection in 3D with a mobile robot*. In: *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, S. 74–80.

[Linder et al., 2015] Linder, T., S. Wehner und K. O. Arras (2015). *Real-time full-body human gender recognition in (RGB)-D data*. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, S. 3039–3045.

[Lowe, 2004] Lowe, David G. (2004). *Distinctive Image Features from Scale-Invariant Keypoints*. Int. J. Comput. Vision, 60(2):91–110.

[Magnusson et al., 2007] Magnusson, Martin, A. Lilienthal und T. Duckett (2007). *Scan registration for autonomous mining vehicles using 3D-NDT*. Journal of Field Robotics, S. 803–827.

[Martinson und Yalla, 2016] Martinson, E. und V. Yalla (2016). *Augmenting deep convolutional neural networks with depth-based layered detection for human detection*. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, S. 1073–1078.

[Maturana und Scherer, 2015] Maturana, D. und S. Scherer (2015). *VoxNet: A 3D Convolutional Neural Network for real-time object recognition*. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, S. 922–928.

[Mitzel und Leibe, 2012] Mitzel, Dennis und B. Leibe (2012). *Taking Mobile Multi-object Tracking to the Next Level: People, Unknown Objects, and Carried Items*. In: *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, ECCV'12, S. 566–579, Berlin, Heidelberg. Springer-Verlag.

[Munaro et al., 2016] Munaro, Matteo, C. Lewis, D. Chambers, P. Hvass und E. Menegatti (2016). *RGB-D Human Detection and Tracking for Industrial Environments*, S. 1655–1668. Springer International Publishing.

[Munaro und Menegatti, 2014] Munaro, Matteo und E. Menegatti (2014). *Fast RGB-D People Tracking for Service Robots*. Auton. Robots, 37(3):227–242.

[Navarro-Sement et al., 2010] Navarro-Sement, Luis Ernesto, C. Mertz und M. Hebert (2010). *Pedestrian Detection and Tracking Using Three-dimensional LADAR Data*. The International Journal of Robotics Research, Special Issue on the Seventh International Conference on Field and Service Robots, S. 1516–1528.

[opencv, 2017] opencv (2017). *Open Source Computer Vision Library*. `https://github.com/opencv/opencv`.

[Pilu et al., 1996] Pilu, Maurizio, A. Fitzgibbon und R. Fisher (1996). *Ellipse-specific direct least-square fitting*. In: *International Conference on Image Processing*, S. 599–602 vol. 3.

[Rusu et al., 2009] Rusu, R. B., N. Blodow und M. Beetz (2009). *Fast Point Feature Histograms (FPFH) for 3D registration*. In: *2009 IEEE International Conference on Robotics and Automation*, S. 3212–3217.

[Rusu et al., 2008] Rusu, R. B., N. Blodow, Z. C. Marton und M. Beetz (2008). *Aligning point cloud views using persistent feature histograms*. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 3384–3391.

[Rusu et al., 2010] Rusu, R. B., G. Bradski, R. Thibaux und J. Hsu (2010). *Fast 3D recognition and pose using the Viewpoint Feature Histogram*. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 2155–2162.

[Rusu, 2009] Rusu, Radu Bogdan (2009). *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. Doktorarbeit, Computer Science department, Technische Universitaet Muenchen, Germany.

[Rusu und Cousins, 2011] Rusu, Radu Bogdan und S. Cousins (2011). *3D is here: Point Cloud Library (PCL)*. In: *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China.

[Schmiedel et al., 2015] Schmiedel, T., E. Einhorn und H. M. Gross (2015). *IRON: A fast interest point descriptor for robust NDT-map matching and its application to robot localization*. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, S. 3144–3151.

[Schneemann, 2013] Schneemann, Friederike (2013). *Recherche und Evaluation von Features zur Detektion von gestürzten Personen in häuslichen Umgebungen*.

[Spinello und Arras, 2011] Spinello, L. und K. O. Arras (2011). *People detection in RGB-D data*. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 3838–3843.

[Spinello et al., 2010] Spinello, Luciano, K. O. Arras, R. Triebel und R. Siegwart (2010). *A Layered Approach to People Detection in 3D Range Data*. In: *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, S. 1625–1630. AAAI Press.

[Sudowe und Leibe, 2011] Sudowe, P. und B. Leibe (2011). *Efficient Use of Geometric Constraints for Sliding-Window Object Detection in Video*. In: *International Conference on Computer Vision Systems (ICVS'11)*.

[Sun et al., 2016] Sun, Y., L. Sun und J. Liu (2016). *Real-time and fast RGB-D based people detection and tracking for service robots*. In: *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, S. 1514–1519.

[Tan und Triggs, 2010] Tan, X. und B. Triggs (2010). *Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions*. IEEE Transactions on Image Processing, 19(6):1635–1650.

[Tombari et al., 2011] Tombari, F., S. Salti und L. D. Stefano (2011). *A combined texture-shape descriptor for enhanced 3D feature matching*. In: *2011 18th IEEE International Conference on Image Processing*, S. 809–812.

[Tombari et al., 2010] Tombari, Federico, S. Salti und L. Di Stefano (2010). *Unique Signatures of Histograms for Local Surface Description*. In: *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, ECCV'10, S. 356–369, Berlin, Heidelberg. Springer-Verlag.

[Vedaldi und Soatto, 2008] Vedaldi, Andrea und S. Soatto (2008). *Quick shift and kernel methods for mode seeking*. In: *In European Conference on Computer Vision, volume IV*, S. 705–718.

[Viola und Jones, 2001] Viola, Paul und M. Jones (2001). *Rapid object detection using a boosted cascade of simple features*. IEEE Conference on Computer Vision and Pattern Recognition, 1(3):I–511 – I–518 vol.1.

[Weinrich et al., 2014] Weinrich, Christoph, T. Wengefeld, M. Volkhardt, A. Scheidig und H.-M. Gross (2014). *Generic Distance-Invariant Features for Detecting People with Walking Aid in 2D Laser Range Data*. In: *Int. Conf. on Intelligent Autonomous Systems (IAS)*.

[Wengefeld et al., 2016] Wengefeld, Tim, M. Eisenbach, Th. Q. Trinh und H.-M. Gross (2016). *May I be your Personal Coach? Bringing Together Person Tracking and Visual Re-identification on a Mobile Robot*. In: *Int. Symposium on Robotics (ISR)*, S. 141–148. VDE.

[Wu et al., 2011] Wu, Shengyin, S. Yu und W. Chen (2011). *An attempt to pedestrian detection in depth images*. In: *2011 Third Chinese Conference on Intelligent Visual Surveillance*, S. 97–100.

[Yu et al., 2012] Yu, S., S. Wu und L. Wang (2012). *SLTP: A Fast Descriptor for People Detection in Depth Images*. In: *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, S. 43–47.