|  |  |  |
| --- | --- | --- |
|  | |  |
| Konzeption und Realisierung eines Personenzählsystems mit Hilfe eines Radarsensors | | |
| Studienarbeit T3200 | | |
|  | | |
| des Studiengangs Elektrotechnik an der Dualen Hochschule Baden-Württemberg Stuttgart | | |
| von  Joshua Rost und Thilo Windemuth | | |
| Abgabedatum: 10.06.2022 | | |
| Bearbeitungszeitraum | 24.01.2022 - 10.06.2022 | |
| Matrikelnummer, Kurs | 7081869, 3873765, TEL19GR5 | |
| Ausbildungsfirma | Infineon Technologies AG, Neubiberg | |
| Betreuer der Dualen Hochschule | Dr. Patrick Köberle | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Erklärung | | | | | |
| Ich versichere hiermit, dass ich meine Studienarbeit T3200 mit dem Thema Konzeption und Realisierung eines Personenzählsystems mit Hilfe eines Radarsensors selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.  Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt. | | | | | |
|  |  |  |  |  |  |
| Ort |  | Datum |  | Unterschrift |  |

Kurzreferat

In Anbetracht der pandemischen Lage durch COVID-19 soll die Anzahl der im Raum befindlichen Personen ermittelt werden. Diese Zählung soll anonym, berührungslos und automatisch funktionieren. Hierzu wird ein 24-GHz Radar der Firma Infineon Technologies AG genutzt.

Es wurde ein Algorithmus entwickelt, der anhand der rohen Radardaten eines Position2Go Radar Demo Kits die Personenzählung durchführt. Hierzu werden zunächst die Rohdaten mithilfe eines Matlabskriptes ausgelesen und in Positions-, Winkel- und Geschwindigkeitsdaten umgewandelt. Die so gewonnenen Daten müssen nun aufbereitet werden, um Messungenauigkeiten zu eliminieren und die Zuverlässigkeit der Objekterkennung zu steigern.

Zuletzt wertet eine Software aus, wie sich die im Durchgangsbereich erkannten Personen bewegen. Dies ermöglicht es dem Programm, die Anzahl der, hinter dem Durchgangsbereich befindlichen, Personen zu überwachen.

Abstract

Considering the pandemic situation due to COVID-19, the number of people in a room shall be tracked. This count should function anonymously, contact-free and automatically. For this purpose, a 24-GHz radar from Infineon Technologies AG is used.

An algorithm was developed, that uses the raw radar data from a Position2Go Radar Demo Kit to perform the people count. For this purpose, the raw data is first read out using a Matlab script and converted into position, angle and velocity data. The data obtained in this way must now be processed to eliminate measurement inaccuracies and increase the reliability of object detection.

Finally, a software tracks the movement of people detected in the passage area. This enables the program to monitor the number of people behind the passage area.

Inhalt

[Abbildungsverzeichnis VI](#_Toc105686196)

[Tabellen VI](#_Toc105686197)

[Abkürzungen VII](#_Toc105686198)

[Formelgrößen und Einheiten VII](#_Toc105686199)

[6.5.1 Messaufbau 19](#_Toc105686200)

[6.5.2 Zuverlässigkeit 20](#_Toc105686201)

[6.5.3 Genauigkeit 21](#_Toc105686202)

[6.5.4 Artefakte 22](#_Toc105686203)

[Literaturverzeichnis 25](#_Toc105686204)

[Anhang 26](#_Toc105686205)

Abbildungsverzeichnis

[Abbildung 1: Einweglichtschranke 4](file:///C:\Users\WindemTh\Desktop\Studienarbeit\Arbeit2\gemeinsameArbeit.docx#_Toc105686206)

[Abbildung 2: Reflektionslichschranke 4](file:///C:\Users\WindemTh\Desktop\Studienarbeit\Arbeit2\gemeinsameArbeit.docx#_Toc105686207)

[Abbildung 3: Personenerkennungssystem mit Kamera 5](file:///C:\Users\WindemTh\Desktop\Studienarbeit\Arbeit2\gemeinsameArbeit.docx#_Toc105686208)

[Abbildung 4: Rückseite mit Antennen 7](file:///C:\Users\WindemTh\Desktop\Studienarbeit\Arbeit2\gemeinsameArbeit.docx#_Toc105686209)

[Abbildung 5: Messergebnis 12](#_Toc105686210)

[Abbildung 6: Ergebnis Datenauswertung 14](#_Toc105686211)

[Abbildung 7: Klasse Target 16](file:///C:\Users\WindemTh\Desktop\Studienarbeit\Arbeit2\gemeinsameArbeit.docx#_Toc105686212)

[Abbildung 8: Sensorbefestigung 19](file:///C:\Users\WindemTh\Desktop\Studienarbeit\Arbeit2\gemeinsameArbeit.docx#_Toc105686213)

[Abbildung 9: Messbereich 19](file:///C:\Users\WindemTh\Desktop\Studienarbeit\Arbeit2\gemeinsameArbeit.docx#_Toc105686214)

Tabellen

[Tabelle 1: Ergebnisse Genauigkeit 21](#_Toc105686215)

[Tabelle 2: Versuchaufbau 22](#_Toc105686216)

[Tabelle 3: Messergebnis Artefakte 22](#_Toc105686217)

[Tabelle 4: Ergebnis nach Filter 23](#_Toc105686218)

Abkürzungen

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Formelgrößen und Einheiten

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

# Einleitung

Das 21. Jahrhundert steht ganz unter dem Zeichen der Digitalisierung. Sensoren und und Mikrocontroller finden Einzug in das tägliche Leben. Die smarten Unterstützer mit ihren digitalen Sinnen finden sich fast überall. Temperatursensoren regeln die Heizungen, Smartwatches erfassen den Puls und Bewegungssensoren schalten das Licht an. Und dieser Trend ist noch lange nicht vorbei, im Gegenteil er scheint gerade erst richtig anzulaufen. In einer Zukunft mit einem Internet der Dinge (IoT), werden wohl noch viel mehr Alltagsgegenstände ihre Umgebung wahrnehmen und die Daten weitervermitteln können [1]. Eine hierbei weitverbreitete Technologie ist das Radar. Radarsysteme sind heutzutage in sehr vielen Bereichen verbreitet. Radar, ein Akronym für ”RAdio Detection And Ranging”, steht wörtlich für ”Ortung und Abstandsmessung durch elektromagnetische Wellen“. Ursprünglich wurden Radare zur Ortung in der Schifffahrt und später in der Luftfahrt entwickelt. Durch die militärische Nutzung, vor allem im 2. Weltkrieg, ist die Forschung und Entwicklung von Radarsystemen weit fortgeschritten [2].

Heutige moderne Anwendungsbereiche von Radargeräten neben militärischen Zwecken beinhalten unter anderem Lichter im Smart Home, die auf bestimmte Gesten und Bewegungen reagieren, oder die Umgebungsuberwachung von Automobilen, wobei bei diesen Radar meist mit anderen Systemen wie Lidar oder Ultraschall kombiniert wird. Zudem werden Radarsysteme bei der Füllstandsmessung oder beim Wetterradar verwendet [3].

In Anbetracht der pandemischen Lage durch COVID-19 soll die Anzahl der im Raum befindlichen Personen ermittelt werden. Um das Infektionsrisiko zu senken, sollen bei der gleichzeitigen Nutzung eines Raums durch mehrere Personen die Anzahl beschränkt werden, sodass eine gewisse Mindestfläche pro Person gewährleistet ist. Diese Zählung der anwesenden Personen soll anonym, berührungslos und automatisch funktionieren [4].

Im Folgenden wird die Struktur der Arbeit erläutert. Zuerst werden die technischen Grundlagen der genutzten Radar Messung erklärt. Als nächstes wird der aktuelle Stand der Technik im Bereich der Personenortung, -aufzeichnung und Datenverarbeitung mit Radar dargelegt. Danach wird im Rahmen einer technischen Machbarkeitsprüfung die Genauigkeit der Winkelmessung des erworbenen Radarsystems überprüft. Zuletzt folgt eine Zusammenfassung der Ergebnisse.

# Grundlagen

# Stand der Technik

Es ist eine Methode gefordert, mit der mehrere Menschen also solche erkannt und gezählt werden können. Außerdem muss die Bewegungsrichtung erfasst werden. Dies soll mit Sensoren geschehen. Schon heute gibt es zahlreiche Personenzählsysteme.

Das wohl verbreitetste und einfachste Vorgehen ist der Einsatz von Lichtschranken.

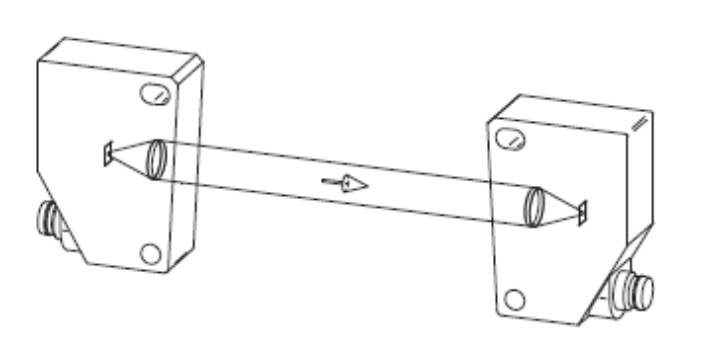
 Meist wird eine Einweglichtschranke verwendet (Abbildung 2). Ein Sender emittiert Licht mittels einer LED. Bei sichtbarem Licht beträgt die Wellenlänge dabei 660 nm. Häufig wird auf Infrarot Strahlung zurückgegriffen, welche für das menschliche Auge nicht sichbar ist. Hier beträgt die Wellenlänge dann 880 – 940 nm. Für hochgenaue Messungen werden Laserdioden verwendet. Der Sender kann dieses Licht mithilfe eine Photodiode oder eines Phototransistors detektieren. Wenn sich nun ein Objekt zwischen Sender und Empfänger befindet, ist die Lichtverbindung zwischen den beiden unterbrochen und der Empfänger detektiert kein Licht mehr.

Abbildung : Einweglichtschranke

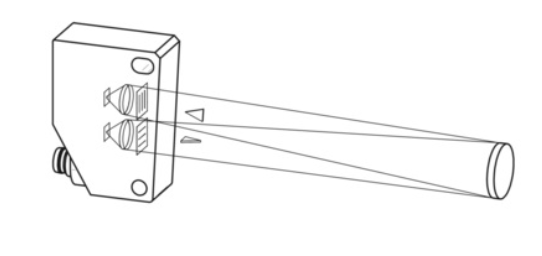
Alternativ gibt es auch die Reflexions-Lichschranke (Abbildung 1). Bei dieser befinden sich Empfänger und Sender im selben Gehäuse. Auf der gegenüberliegenden Seite befinden sich ein Reflektor, welcher das Licht zurück wirft [5].

Abbildung : Reflektionslichschranke

Es gibt nun mehrere Möglichkeiten, mit Hilfe von Lichtschranken ein Personenzählsystem zu realisieren. Eine ist es, einen festen Ausgang und einen festen Eingang zu definieren und an beiden eine Lichtschranke zu positionieren. Jede Unterbrechung der Lichtschrank kann dann als eintretender oder austretender Mensch gewertet werden. Es kann zu Fehlern kommen, wenn Menschen nebeneinander gehen, oder sich nicht an Eingang und Ausgang halten.

Eine andere Möglichkeit ist es, ein Lichtgitter zu errichten. Dies sind mehrere Lichtschranken hintereinander. Wenn ein Mensch dieses Lichtgitter durchquert, unterbricht er nacheinander die verschiedenen Lichtschranken. So kann die Laufrichtung bestimmt werden, und es sind keine getrennten Ein- und Ausgänge mehr notwendig. Allerdings ist die Installation aufwendig und nebeneinander eintretende Menschen können zu Fehlern führen [6].

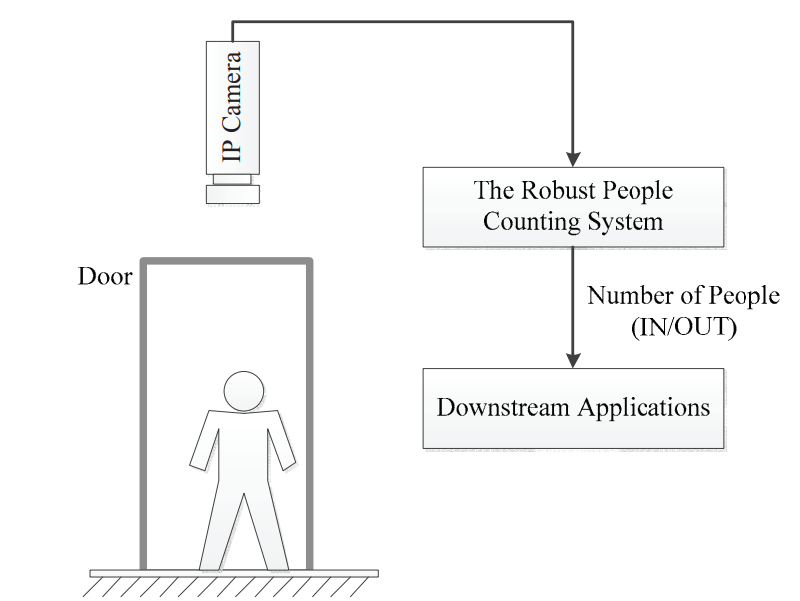
Eine andere häufig eingesetze Technologie sind Kameras. Der schematische Aufbau eines solchen Personenzählsystems ist in Abbildung 3 dargestellt.

Abbildung : Personenerkennungssystem mit Kamera

Die Kamera befindet sich überhalb des Eingangs und filmt die Menschen von oben.

Ein Bildverarbeitungsalgorithmus, welcher Menschen erkennen kann, wird nun auf die einzelnen Bilder angewandt. Die den Kamerabereich durchquerende Person ist also auf mehreren aufeinanderfolgenden Bilder an ihrer jeweiligen Position detektierbar. Über diese Bilderstrecke lässt sich herausfinden, in welche Richtung die Person sich bewegt. Mit Hilfe dieser Daten kann ein Personenzählalgorithmus realisiert werden. Mehrere Personen können gleichzeitig erfasst werden [7, 8].

Auch beim autonomen Fahren müssen Menschen zuverlässig erkannt, lokalisiert und ihre Bewegungsrichtung erfasst werden. Hier konkurrieren zwei Technologien miteinander. LiDAR (Light Detection and Range) und RADAR (Radio Detection and Ranging). Die Funktionsweise der beiden Technologien ist sehr ähnlich. Strahlung wird emittiert und die Reflektion gemessen. Mit diesen Daten lassen sich Objekte erkennen, lokalisieren und die Geschwindigkeit bestimmen.

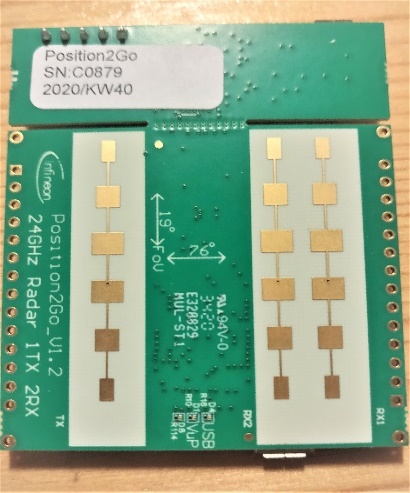
Der Unterschied ist, dass beim Radar elektromagntische Wellen emittiert werden während LiDAR mit Licht arbeitet [9].

LiDAR Sensoren, wie sie von vielen großen Autobauern, wie zum Beispiel Volkswagen oder Toyota, eingesetzt werden, sind sehr genau. Sie können die Umgebung bis auf wenige Zentimeter genau erfassen. Dafür ist ihre Reichweite vergleichsweise gering und der Preis sehr hoch. Bei schwieriger Witterung, wie zum Beispiel bei Nebel haben LiDAR Sensoren allerdings Schwierigkeiten [10].

RADAR-Sensor sind deutlich billiger als LiDAR Sensoren. Außerdem sind sie unempfindlich gegenüber dem Wetter. Elektromagnetische Strahlung wird von Schnee und Regen kaum beeinflusst. Die Technologie birgt außerdem den Vorteil, die Radialgeschwindigkeit eines Objektes mithilfe des Dopplereffektes bestimmen zu können. Der wohl bekannteste Elektroautohersteller Tesla, setzt aus diesen Gründen bei der Erkennung des Straßenverkehrs für Fahrerassistenzsysteme, auf Radar [11].

Bei der Erkennung von Personen mit Radarsystem werden meist entweder sogenannte Impulse-Radio Ultra-Wideband (IR-UWB)-Radarsystem, also mit gepulsten breitbandigen elektromagnetische Wellen [12] [13] [14], verwendet oder Systeme mit FMCW-Radar [3] [15] [16]. Diese IR-UWB-Systeme haben eine höhere Auflösungsgenauigkeit, jedoch geringe DopplerInformationen [16]. Allerdings können beide Systeme mit einer hohen Genauigkeit bis zu 4 Personen zählen (95,8% bei Choi et al. [14] und 85,4% bei Weiss et al. [16]) Auch in dieser Studienarbeit wird das Problem mit Radarsensoren angegangen.

# Komponenten

Das „Position2Go Development Kit“ besteht aus einem Board, auf dem folgende Komponenten aufgelötet sind:

* BGT24MTR12:  
  Der 24GHz Radarchip besitzt eine sendende und zwei empfangende Antennen.

In Abbildung 4 ist die Rückseite des Sensor abgebildet. Auf der linken Seite befindet sich die sendende Antenne und auf der rechten die beiden empfangenden Antennen.

Abbildung : Rückseite mit Antennen

* XMC4700 Cortex-M4:

Dieser Mikrocontroller kann genutzt werden, um die Rohdaten direkt zu verarbeiten

* XMC4200:

Dies ist ein onboard Debugger mit einer UART Schnittstelle. Das Debuggerboard lässt sich bei Bedarf physisch abbrechen [17].

# Sensordatenverarbeitung

Für die Datenverarbeitung und rudimentäre Personenerkennung liefert Infineon Beispielscode. Dieser arbeitet nicht mit echten Sensordaten, sondern mit einem beispielhaften Datenset, welches in einer XML Datei gespeichert wird. Mit dieser lassen sich dann die Daten von einem Frame ausgewertet werden. Teile dieses Beispielcodes können verwendet werden. Um allerdings mit echten Sensordaten zu funktionieren, muss zunächst der Sensor richtig eingestellt werden. Dann müssen die echten Daten und auch die aktuellen Einstellungen aus dem Sensor ausgelesen werden. Die Objekterkennung kann mit leichter Modifikation übernommen werden.

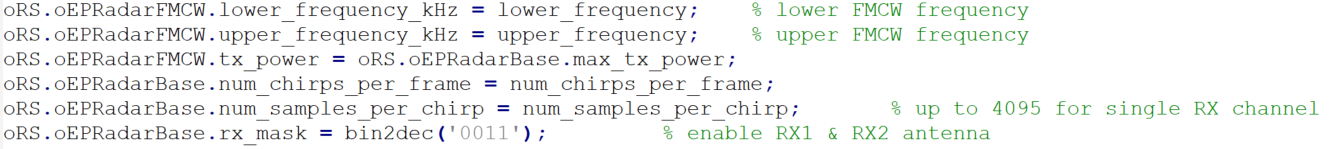
## Einstellung des Sensors

Zunächst muss der Sensor mit den richtigen Einstellungsparametern versorgt werden. Der Radarsensor ist wie in Listing 1 gezeigt als Objekt oRS der Klasse RadarSystem() initialisiert. Der richtige Port wird dabei als Eingabeparameter mitgegeben



Listing 1: Initialisierung oRS

Die Verschiedenen Einstellungen sind nun Attribute des Objektes oRS. Diese werden, wie in Listing 2 gezeigt, gesetzt.



Listing 2: Radarsensor Einstellungen

Zunächst wird das Frequenzband festgelegt, in dem der Sensor sendet. Wie im UserManual empfohlen, werden die Frequenzen zwischen 24,05 GHz und 24,22 GHz genutzt [18].

Das Attribut tx\_power beschreibt die Sendeleistung des Sensors. Diese wird auf den Maximalwert gesetzt um ein bestmögliches Ergebnis zu erzielen. Der Maximalwert kann direkt aus dem Sensor ausgelesen werden [18].

Die Anzahl der chirps in einem Frame wird auf die maximael Anzahl 16 gesetzt. Hiermit wird die höchste Auflösung erreicht [18].

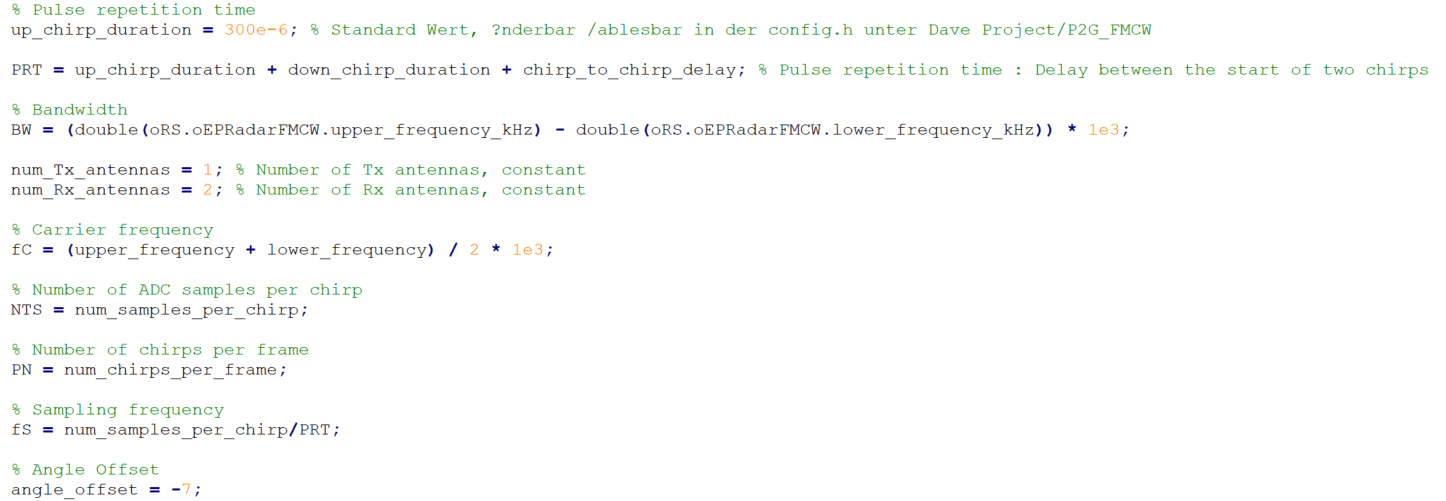
Die Anzahl der samples\_per\_Chirp also der Messwerte pro Chirp wird auf 128 gesetzt. Der höchst möglich Wert wäre 256. Dieser wurde nicht genommen, um die Rechenzeit zu reduziere, die pro Chirp gebraucht wird. Dies ist insofern wichtig, da viele Chirps eingestellt wurde und die Geschwindigkeit der Auswertung reduziert werden soll [18].

Über rx\_mask kann dem Chip mitgeteilt werden, welche Antennen genutzt werden sollen. In diesem Fall werden beide Antennen RX1 und RX2 als Empfangsantennen genutzt [18].

## Daten Auslesen

Um die Daten eines einzelnen Frames auszulesen, bietet die mitgelieferte Matlab Bibliothek die Methode get\_frame\_data. Diese gibt den Struct sInfo und die Matrix mxData zurück. sInfo enthält Informationen über den Frame und mxData enthält die eigentlichen Daten von den beiden empfangenden und der sendenden Antenne [18].

Für die weitere Verarbeitung werden noch weitere Daten benötigt. In der Bespielauswertung werden diese aus der XML-Datei ausgelesen. Für die Auswertung müssen sie erst festgelegt werden. Listing 3 zeigt die Definition weiterer für die Messwertauswertung wichtiger Paramter. Hierbei wurden bereits vorhandene Werte zum berechnen genutzt. Die Kommentare beschreiben jeweils die Bedeutung der Werte. Der angle\_offset wurde experimentell bestimmt. Alle vom Sensor berechneten Winkel sind versetzt und nicht korrekt. Der Offset muss addiert werden um die richtigen Werte zu erhalten.

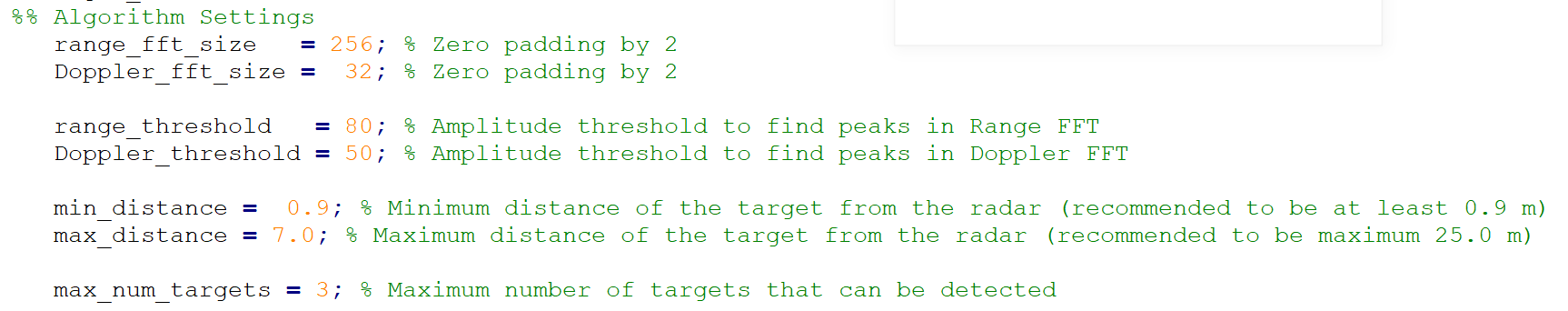


Listing 3: Messeinstellungen

## Datenverarbeitung

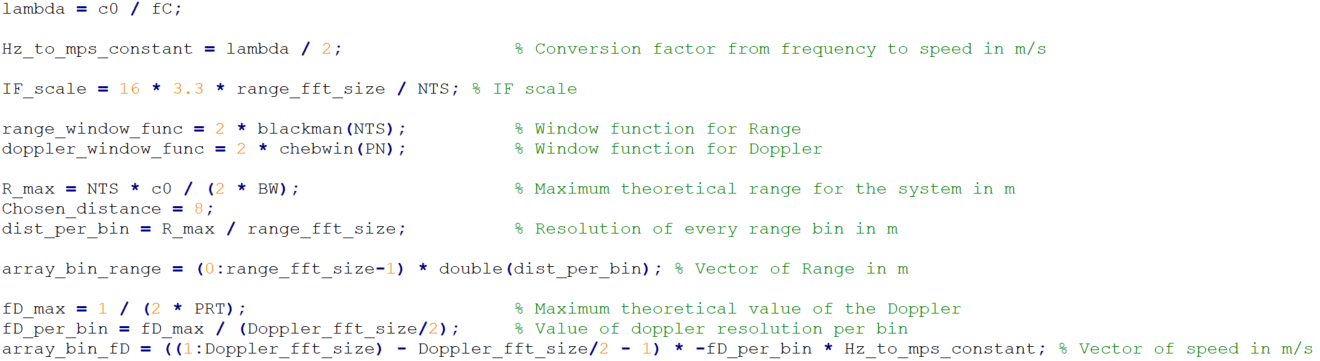
Die Datenverarbeitung funktioniert nach dem im Kapitel Grundlagen definierten Schema. Die Implementation wurde grundsätzlich aus dem vorhanden Beispielcode übernommen. Es wurden allerdings eigene Einstellungen vorgenommen und kleinere Fehler behoben.

Zunächst wurden die Algorithmuseinstellungen definiert. Die Auflösung der Fast Fourier Transformation ist dabei in Kompromiss zwischen Genauigkeit und Rechenaufwand. Der Threshold markiert jeweils das Amplitudenlevel, ab dem bei einer bestimmten Frequenz eine Person erkannt wird. Dieser ist in der Standardeinstellung für die Distanz bei 50. Um Artefakte zu vermeiden wurde die range\_threshold auf 80 erhöht. Da die Sendeleistung auf das Maximum gesetzt wurde, können Menschen hier immer noch erkannt werden. Experimente bei einer einem Wert von 50 haben sehr schlechte Ergebnisse geliefert. Viele Artefakte haben diese Grenze überschritten. Außerdem funktioniert in einem solchen Fall die Maximumsuche nicht mehr richtig und verschiedene Personen werden von dem Algorithmus vermischt. Die maximale Anzahl an gleichzeitg erkannten Personen wurde auf 3 gesetzt. Der Messbereich ist zwischen 0.9 und 7 Metern gewählt. Bei größeren Distanzen zu Sensor leidet die Genauigkeit.



Listing 4: Algorithmuseinstellungen

Im Folgenden werden weitere für die Auswertung wichtige Parameter berechnet. Dies ist in Listing 5 dargestellt.



Listing 5: Berechnete Parameter

Zu Beginn der Auswertung wird der Analog Digital Wandler (ADC) kalibriert. Dann werden die Sensordaten verarbeitet und Fouriertransformiert. Das Ergebnis ist für jede Empfangsantenne eine Ergebnismatrix, die Signalamplitude bei der jeweiligen Frequenz enthält. Abbildung 5 zeigt ein visualisiertes Beispiel. Hier ist in blau Antenne Rx1 und in grün Antenne Rx2 dargestellt. In diesem Bespiel ist die Grenze range\_threshold auf 40 eingestellt.

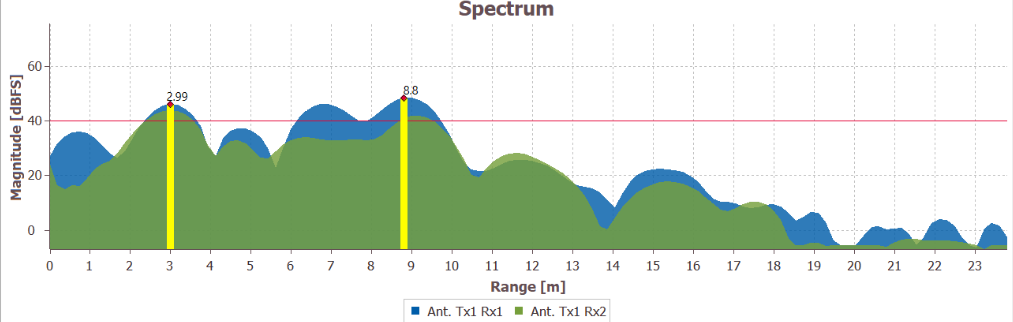


Abbildung 5: Messergebnis

Auf diese Daten wird dann pro Antenne die Methode f\_search\_peak angewandt. Diese sucht in den Frequenzbereichen, die über range\_threshold liegen nach dem Amplitudenmaximum. An diesem wird dann eine Person erkannt. Bei Abbildung 5 ist dies bei 2.99 und bei 5 m der Fall. Die X-Achse ist in diesem Fall in Meter umgerechnet, könnte aber auch als Frequenz dargestellt werden.

## Optimierung

Um regelmäßig Daten auszulesen, wird der Auswertungsalgorithmus in einer „while“-Schleife immer wieder ausgeführt. Dies funktionierte zunächst nur sehr langsam (~1 Wert/s). Um die Geschwindigkeit zu erhöhen wurde alle Codeteile, die nur einmal nötig sind aus der Schleife rausgenommen. Hierzu gehört die Initialisierung aller Variablen und Objekte, das einmalige Auslesen von Sensoreinstellungen, das einstellen des Sensors, die Berechnung der Konstanten, die Algorithmuseinstellungen und die Kalibrierung des Analog Digital Konverters. Nach dieser Anpassung läuft der Code deutlich schneller und schafft 20 Werte/s.

## Ergebnis

Um das Ergebnis zu prüfen und analysieren wurden die Ergebnisdaten in einem wachsenden Liniendiagram visualisiert. Ein Beispielplot ist in Abbildung 6 gezeigt. Zunächst lässt sich erkennen, das regelmäßig Daten ausgelesen werden. In dem 2 Diagram wird zum Beispiel die Distanz visualisiert. Hier werden zu Beginn zwei Objekte erkennt. Eines bei etwa 6 und eines bei etwas über 2 Metern. Gegen Ende entfernen sich die beiden Objekte dann. Der Geschwindigkeitsplot liefert relativ schlechte Werte und erkennt meistens eine Geschwindigkeit von 0 m/s. Das Winkel Diagram zeigt keine Werte, da der Winkel meist negativ und somit außerhalb der Diagramdarstellung liegt.

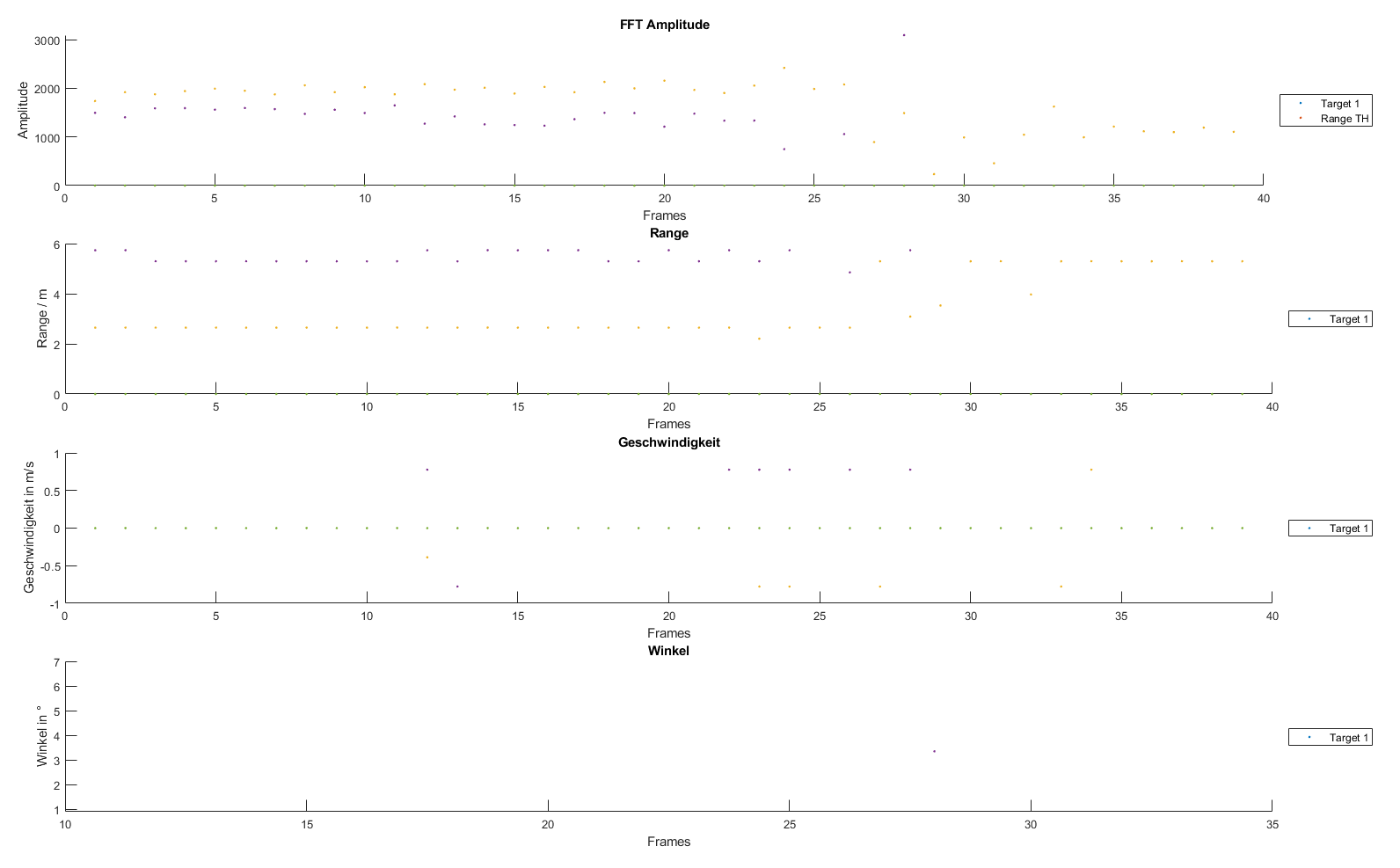


Abbildung 6: Ergebnis Datenauswertung

Die Datenverarbeitung funktioniert im Rahmen der Möglichkeiten. Es werden regelmäßig Daten ausgelesen, eine Objekterkennung durchgeführt und die Werte ausgegeben. Diese Werte können nun weiter verarbeitet werden.

# Durchschnittsdaten bilden

Die Personenerkennung, die mit den aktuellen Sensordaten durchgeführt wird, ist noch fehlerhaft. Wenn sich eine Person in dem überwachten Bereich befindet, wird sie nicht durchgehend erkannt. Häufig gibt es Aussetzer und der Sensor erkennt nichts. Außerdem werden viele Artefakte registriert. Obwohl sich an einer Stelle keine Person befindet, wird dort scheinbar zufällig eine erkannt. Mit diesen Problemen lässt sich ein funktionierender Personenzähler nicht realisieren.

Als Lösung wird der Durchschnitt aus einer bestimmten Anzahl an Messdaten gebildet. Diese Anzahl heißt average\_nr. Jede dieser Messdaten kann eine bestimmte maximale Anzahl an erkannten Personen enthalten. Diese Anzahl ist mit der Variablen max\_num\_targets definiert. Werden weniger Personen erkannt als möglich, werden die freien Stellen mit NaN aufgefüllt. Die Anzahl getrackten Personen, die sich in einem solchen Datenset befinden, ergibt sich aus der Multiplikation von average\_nr und max\_num\_targets. Für jede Person wird die Distanz, der Winkel und die Geschwindigkeit gespeichert.

## Messdaten speichern

Um ein Datenset zu erstellen, müssen die aktuellen Daten gespeichert werden. Die verschiedenen Daten werden jeweils in Feldern mit der Länge average\_nr gespeichert (range(), speed(), angle()). Zu Beginn der Messung werden diese Felder sukzessiv aufgefüllt. Sind die Felder voll, wird bei jeder Messung der erste Wert des Feldes gelöscht. Die verbleibenden Werte rutschen eine Position nach vorne und der neue Messwert wird hinten angehängt. So enthält das Feld immer die gewünschte Anzahl der vergangenen Messwerte.

## Vorgehen

Die erkannten Personen werden alle nacheinander durchgearbeitet. Es gibt die Möglichkeit, eine Person mit ihrer jeweiligen Position zu speichern.

Die Position der aktuellen Person, wird zunächst mit den Position aller bereits gespeicherten Personen verglichen. Wenn sie sich in unmittelbarer Nähe einer schon gespeicherten Person befindet, wird davon ausgegangen, dass es sich um dieselbe Person handelt. Anstatt eine neue Person zu speichern, wird die aktuelle Person zu der schon gespeicherten hinzugefügt. Als Position wird ein Mittelwert der beiden Positionen gebildet. Jede Person speichert außerdem, wie viele Messewerte ihr zugeordnet wurden. Bei 20 Messungen wären das idealerweise 20. Gibt es nun aber Messfehler, sind es weniger. Es wird nun eine Grenze definiert (appearance\_border), die angibt bei wie vielen Messungen eine Person mindestens erkannt werden muss, um berücksichtigt zu werden. Wird eine Person seltener erkannt, wird davon ausgegangen, dass es sich um ein Messartefakt handelt. In diesem Fall werden die Daten nicht beachtet und gelöscht.

Mit diesem Vorgehen lassen sich die Messungenauigkeiten des Sensors ausgleichen und eine korrekte Personenerkennung ist möglich. Je höher hierbei die Anzahl der analysierten Messungen (average\_nr) ist, desto zuverlässiger werden die Daten. Der Zeitraum muss allerdings so gering sein, dass die Person ihre Position nicht ausschlaggebend ändern kann.

## Implementierung

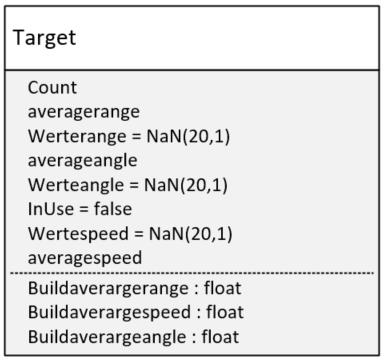
Die Speicherung der Daten einer Person, erfolgt in der in Abbildung 7 dargestellten Klasse Target. Für jede Person wird ein Objekt dieser Klasse erstellt.

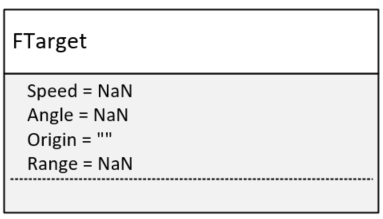
Abbildung : Klasse Target

Die Variable Count speichert, wie viele Messwerte der Person zugeordnet wurden. Eine Person besitzt immer drei Werte: Distanz (range), Winkel (angle) und Geschwindigkeit (speed). Zu diesen wird immer jeweils ein Durchschnitt (zb. averagerange) und ein Feld (zb. Werterangerange) gespeichert. In diesem Feld werden alle einzelnen Messwerte gespeichert, die der Person zugeordnet wurden. Das Feld wird mit der Größe von average\_nr(hier 20) ohne Werte initialisiert.

Außerdem existiert für jeden Wert eine Methode, die den Durchschnitt berechnet (zb. Buildaveragerange). Diese Methode bildet aus den Werten des jeweiligen Feldes den Durchschnitt und gibt diesen als Rückgabewert zurück.

Das Attribut InUse gibt an, ob diese Person schon genutzt wird und wird mit false initialisiert.

Nun wird das in 6.2 beschriebene Vorgehen angewandt. Zu Beginn wird ein Feld mit dem Namen object\_array der Größe max\_objects an Objekten der Klasse Target erstellt. Wird nun eine neue Person gespeichert, wird die erste noch ungenutzte Person des Feldes genutzt (InUse=true). Die jeweiligen Werte werden in das jeweilige Feld eingetragen. Auch das jeweilige Durchschnittsattribut bekommt diesen Wert. Der Count bekommt den Wert 1.

Neue Messwerte können nun mit den in den Objekten gespeicherten Messwerten verglichen werden. Wird ein neuer Messwert einer Person zugeordnet, wird der jeweilige Wert dem jeweiligen Feld angehängt. Danach wird die Methode, welche den Durchschnitt ausrechnet aufgerufen und der Rückgabewert in die Durchschnittsvariable gespeichert. Der Count wird um 1 erhöht.

Am Ende der Durchschnittsberechnung gibt es noch ein zweites Feld an Objekten der Klasse Ftarget erstellt. Es hat den Namen output\_array. In dieses Feld werden nur die Objekte übernommen, die einen Count von mindestens appearance\_border besitzen, also oft genug erkannt wurden. Die Klasse Ftarget besitz als Attribute nur die jeweiligen Messewerte und die Herkunft des Objektes (Origin). Die Herkunft wird erst später wichtig. Wenn eine Person aus object\_array in output\_array übernommen wird, werden jeweils die Durchschnittswerte in Target in die jeweiligen Attribute von Ftarget übernommen.

## Parameterwahl

|  |  |  |
| --- | --- | --- |
| average\_nr | Anzahl der Messungen von denen ein Durchschnitt gebildet wird | 20 |
| Range\_tolerance | Abstandstoleranz bei Zuordnung von Messdaten zu Person in Metern | 1 |
| Angle\_tolerance | Winkeltoleranz bei Zuordnung von Messdaten zu Person in Grad | 17 |
| Appearance\_border | Minimal nötige Anzahl an Messwerten um weiterverarbeitet zu werden | 6 |
| Max\_objects | Maximal mögliche Anzahl an speicherbaren Personen | 20 |

Um die Parameter bestmöglich zu bestimmen wurde zunächst die Geschindigkeit des Codes getestet. Maximal ist es möglich, 20 Messwerte in einer Sekunde zu erfassen. Der Sensor liefert allerdings sehr schlechte Daten. Tests haben ergeben, dass der Durchschnitt von mindestens 20 Daten (average\_nr) benötigt wird, um zuverlässig Personen zu erkennen. Dies entspricht also den Daten einer ganzen Sekunde. Die Toleranzen wurden deswegen sehr großzügig gewählt. Der Winkel wird von dem Sensor besonders ungenau gemessen. Die Winkeltoleranz ist deswegen sehr hoch gewählt. Dies führt dazu, dass nur Personen die einen genügend großen Abstand voneinander haben gut erkannt werden. Dies ist dem Sensor geschuldet.

Auch die Appearance\_border ist relativ klein gewählt. Es reicht, wenn eine Person in 6 von 20 Messungen erkannt wird, um als Person verarbeitet zu werden. Auch dieser Wert wurde durch Tests festgelegt. Wird er noch kleiner gewählt steigt die Zahl der erkannten Artefakte. Wird er allerdings höher gewählt, werden existierende Personen nicht mehr immer erkannt.

Es wird ein laufender Mittelwert gebildet. Das bedeutet es werden zwar die Messwerte einer ganzen Sekunde verarbeitet, allerdings geschieht das 20 mal in der Sekunde. Hierdurch wird die größmöglichste Auflösung erreicht, ohne an Zuverlässigkeit zu verlieren.

## Test

### Messaufbau

Die Freiluftmessung wurde in der in Abbildung 9 abgebildeten Einfahrt durchgeführt. Die Hecke befindet sich in einem Abstand von 8.25 Metern. Der Sensor wurde allerdings so eingestellt, das Objekte nur bis maximal 8 Metern erkannt werden. Abbildung 8 zeigt die Befestigung des Sensors.

Abbildung : Sensorbefestigung



Abbildung : Messbereich

Eine Person stellt sich auf die jeweils ausgemessene Position. Die Messung wird bei 3, 5 und 7 Metern bei jeweils 0° durchgeführt. Um die Verbesserung darzustellen werden die Daten mit identischen Messung verglichen, bei der nur die unverarbeiteten Sensorwerte gemessen wurden. Es werden jeweils 20 Messungen untersucht.

### Zuverlässigkeit

|  |  |
| --- | --- |
| Unverarbeitete Messwerte | Verarbeitete Messwerte |
|  |  |
|  |  |
|  |  |

Man sieht sehr eindrucksvoll, dass der Algorithmus zum bilden eines Durchschnitts sehr gut funktioniert. Die Rohdaten sind sehr unzuverlässig. Bei 5 Metern wird ein vorhandenes Objekt öfters nicht erkannt als das es erkannt wird. Nach der Verarbeitung wird bei allen Distanzen in 100 Prozent der Messungen die Person erkannt. Das Problem der unzuverlässigen Erkennung wurde also gelöst.

### Genauigkeit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Wahrer Wert | 3 m | 5 m | 7 m |
| Rohdaten | Durchschnitt | 3.12 m | 5.17 m | 6.6 m |
| Standartabweichung von Mittelwert | 0.09 m | 0.09 m | 0.2 m |
| Durchschnittliche Abweichung von realem Wert | 0.12 m | 0.18 m | -0.39 m |
| Verarbeitete Daten | Durchschnitt | 3.09 m | 5.13 m | 7.21 m |
| Standartabweichung von Mittelwert | 0.01 m | 0.03 m | 0.07 m |
| Durchschnittliche Abweichung von realem Wert | 0.09 m | 0.13 m | 0.21 m |

Tabelle 1: Ergebnisse Genauigkeit

Bei der Genauigkeit lässt sich nur eine geringfügige Verbesserung feststellen. Die Werte sind immer noch ähnlich genau.

### Artefakte

In einem Versuchsaufbau wurden zwei Personen in Messbereich, wie in Tabelle 2 beschrieben, platziert.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Abstand  in Meter | Winkel  in Grad | Geschwindigkeit  in m/s |
| Person 1 | 5.8 | -15 | 0 |
| Person 2 | 4.1 | 10 | 0 |

Tabelle 2: Versuchaufbau

Nun wird ein Datenset (20 Messwerte) ausgewertet. Das Ergebnis ist in object\_array gespeichert. Der Inhalt ist in der Tabelle 3 dargestellt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Count | averagerange | averageangle | averagespeed |
| Target 1 | 9 | 5.7843 m | -13.0065° | -0.0863 m/s |
| Target 2 | 2 | 1.3235 m | -6.9661° | 0 m/s |
| Target 3 | 9 | 4.0196 m | 8.2395° | 0.2590 m/s |
| Target 4 | 4 | 6.0662 m | 9.6702° | -0.3884 m/s |
| Target 5 | 3 | 5.7353 m | 1.5411° | -0.2590 m/s |

Tabelle 3: Messergebnis Artefakte

Es ist zu erkennen das der Sensor 5 Objekte lokalisiert, obwohl in echt nur 2 vorhanden sind. Die beiden echten Personen sind also Target 1 und 3 gespeichert. Die echten Personen lassen sich von den Artefakten durch den höheren Count unterscheiden.

Nachdem nach Count gefiltert wurde bleiben nur die richtigen in dargestellten Personen übrig.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Count | averagerange | averageangle | averagespeed |
| Target 1 | 9 | 5.7843 m | -13.0065° | -0.0863 m/s |
| Target 3 | 9 | 4.0196 m | 8.2395° | 0.2590 m/s |

Tabelle 4: Ergebnis nach Filter

Das Program ist meist in der Lage, die echten Personen von Artefakten zu unterscheiden. In den meisten Fällen funktioniert dies auch wie in dem hier dargestellten Beispiel. Manchmal allerdings, liefert der Sensor auch über viele Messwerte hinweg denselben falschen Wert. In einem solchen Fall kann ein Artefakt auch bis in die Auswertung gelangen und muss dort anderweitig abgefangen werden. Die Ursache für das Verhalten des Sensors ist hierbei unbekannt.

# Zusammenfassung

Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | D. L., „Wie die Digitalisierung unseren Alltag verändert,“ 20 April 2020. [Online]. Available: https://www.forschung-und-wissen.de/nachrichten/technik/wie-die-digitalisierung-unseren-alltag-veraendert-13373812. [Zugriff am 23 Januar 2022]. |
| [2] | S. Steffen, FMCW-Radarsignalverarbeitung zur Entfernungsmessung mit hoher, Karlsruhe: Karlsruher Institut fur Technologie (KIT), 2017. |
| [3] | V. P. C. A. u. S. A. Will Christoph, „Human target detection, tracking, and classification using 24-ghz fmcw radar,“ *IEEE Sensors Jounal,* Bd. 19, p. 7283–7299, 2019. |
| [4] | „Pandemie der Coronavirus-Krankheit,“ United Nations, [Online]. Available: https://www.euro.who.int/de/health-topics/health-emergencies/coronavirus-covid-19#:~:text=Am%2031.,2019%2DnCoV%E2%80%9C%20bezeichnet%20wurde.. [Zugriff am 23 Januar 2022]. |
| [5] | „Funktionsweise und Technologie von Lichtschranken und Lichttastern,“ 23 January 2021. [Online]. Available: https://www.baumer.com/de/de/service-support/funktionsweise/funktionsweise-und-technologie-von-lichtschranken-und-lichttastern/a/Know-how\_Function\_Lichtschranken-Lichttaster. |
| [6] | „Lichtschranke: Funktion, Arten,“ [Online]. Available: https://wiki.induux.de/Lichtschranken#:~:text=%F0%9F%8E%93%20Die%20Lichtschranke%20ist%20ein,durch%20eine%20elektrisches%20Signal%20aufgezeigt.&text=Der%20Sender%20stellt%20die%20Lichstrahlenquelle,die%20Strahlung%20der%20Lichtquelle%20aufnimmt.. |
| [7] | E. P. Myint, „People Detecting and Counting System,“ in *2021 IEEE 3rd Global Conference on Life Sciences and Technologies*, 2021. |
| [8] | P. Bamrungthai und S. Puengsawad, „Robust People Counting Using a Region-Based Approach for a Monoculat Vision System,“ in *International Conference on Science and Technology*, Samutprakarn, 2015. |
| [9] | A. Neal, „Fierce Electronics,“ 24 April 2018. [Online]. Available: https://www.fierceelectronics.com/components/lidar-vs-radar. |
| [10] | E. Voigt und M. Kremsreiter, *LiDAR in Anwendung,* Chemnitz: TU-Chemnitz, 2020. |
| [11] | „Radar - technische Erläuterung,“ [Online]. Available: https://www.ramom.de/rechtsthemen/radarfalle/radar-technische-erlaeuterung.html. |
| [12] | Q. X. C. S. H. Choi Jeong Woo, „Bi-directional passing people,“ *IEEE Internet of Things Journal,* p. 512–522, 2018. |
| [13] | C. J.-H. K. K.-T. Kim Ji-Eun, „Robust detection of presence of individuals in an indoor environment using ir-uwb radar,“ IEEE, 2020. |
| [14] | Y. D. H. C. S. H. Choi Jeong Woo, „People counting based on an,“ *IEEE Sensors Journal,* p. 5717–5727, 2017. |
| [15] | H. S. S. A. W. R. F. G. Stephan Michael, „ People,“ *IEEE Sensors,* p. 1–4, 2021. |
| [16] | P. R. E. Weiss Jonas, „ Improved people counting algorithm,“ *IEEE Radar Conference,* pp. 1-6, 2020. |
| [17] | *24 GHz transceiver - BGT24MTR12,* München: Infineon Technologies AG, 2020. |
| [18] | *Position2Go software user manual,* München: Infineon Technologies AG, 2019. |

Anhang