

Julius-Maximilians-Universität Würzburg

Fakultät für Mathematik und Informatik

Informationstechnik für Luft- und Raumfahrt

Lehrstuhl für Informatik 8

Prof. Dr. Sergio Montenegro



Bachelorarbeit

Intelligentes Mapping für Indoor-Quadrocopter

Vorgelegt von

Norbert Schmitt

Matr.-Nr.: 1716798

Prüfer: Prof. Dr. Sergio Montenegro

Betreuender wissenschaftlicher Mitarbeiter: Dipl.-Ing. Nils Gageik

Würzburg, 08. 12. 2012

Erklärung

Ich versichere, dass ich die vorliegende Arbeit einschließlich aller beigelegter Materialien selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Werken entnommen sind, sind in jedem Einzelfall unter Angabe der Quelle deutlich als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form noch nicht als Prüfungsarbeit eingereicht worden.

Mir ist bekannt, dass Zuwiderhandlungen gegen diese Erklärung und bewusste Täuschungen die Benotung der Arbeit mit der Note 5.0 zur Folge haben kann.

Würzburg, 08. 12. 2012

Vorname Nachname

Aufgabenstellung

Die Fortschritte im Bereich Sensorik und Mikrotechnik ermöglichen heutzutage den kostengünstigen Bau kleiner unbemannter Luftfahrzeuge (UAV, unmanned aerial vehicle, Drohne) wie Quadrocopter. Die Forschung und Entwicklung dieser Systeme wurde in den letzten Jahren aufgrund der vielfältigen Anwendungsmöglichkeiten stark vorangetrieben. Wenngleich im Bereich UAV viel geforscht wurde, ist das Thema Autonomes Flugobjekt längst noch nicht vollständig behandelt. Insbesondere der Indoor-Betrieb ist aufgrund fehlender absoluter Positionsstützung durch GPS problematisch. Der Aufbau eines eigenen autonomen Systems wird daher am Lehrstuhl Aerospace Information Technology der Uni Würzburg erforscht und erprobt. Im Rahmen dieses Forschungsvorhabens ist ein System zu entwickeln, dass in der Lage ist Sensordaten seiner Umgebung zu einer Repräsentation seiner Umwelt (Mapping) zu verbinden.

Hauptaugenmerk dieser Arbeit ist die Erstellung einer Übersicht von Mapping Lösungen. Die einzelnen Lösungsmöglichkeiten sind zu beschreiben sowie Vor- und Nachteile gegeneinander abzuwägen. Schließlich ist eine Lösung unter Berücksichtigung der zur Verfügung stehenden Sensorik (Ultraschall und Infrarot) zu implementieren. Die Arbeit ist umfangreich zu evaluieren und ausführlich zu dokumentieren.

Aufgabenstellung (Stichpunktartig):

- Ausführliche Recherche: Mapping Lösungen
- Implementierung einer Mapping Lösung
- Einbinden in Quadrocopter-Code
- Evaluierung
- Dokumentation

Zusammenfassung

Diese Arbeit beschäftigt sich mit Mapping-Lösungen, die für den Einsatz in Quadrocoptern geeignet sind. Dabei werden gängige Verfahren vorgestellt und deren Vor- und Nachteile gegeneinander abgewogen. Es wurden zwei geeignete Verfahren zum Kartografieren der Umgebung gewählt und als Mapping-Lösung implementiert. Das Konzept berücksichtigt dabei die Besonderheiten der im Quadrocopter verbauten Ultraschallsensoren. Die leichten und kostengünstigen Ultraschallsensoren besitzen einen großen Öffnungswinkel, wodurch sich große Bereiche mit wenigen Sensoren abdecken lassen. Sie funktionieren auch unter schlechten Bedingungen, wie Rauch, in denen optische Sensoren starke Probleme haben. Die Eigenschaften der Sensorik werden über ein Modell im Computer abgebildet. Einfache Line-Following-Algorithmen liefern jedoch nur unzureichende Ergebnisse, weswegen ein sogenanntes Sektormodell verwendet wird. Auch wird das Signal des Ultraschallsensors bei zu flachem Einfallswinkel in die falsche Richtung reflektiert und es kommt zu fehlerhaften Messungen. Für das Mapping wurde ein Bayes-Filter (Bayes-Verfahren) sowie ein Zählverfahren implementiert. Die Implementierung wurde einerseits in die Steuersoftware des Quadrocopter und andererseits auf dem PC in die Software der Bodenstation integriert. Die beiden Verfahren wurden gegenübergestellt und die Parameter optimiert.

Inhaltsverzeichnis

1. Einleitung	1
2. Stand der Technik	2
2.1. Mapdarstellung	3
2.1.1. Geometrische Darstellung	3
2.1.2. Topologische Darstellung	4
2.1.3. Grid Darstellung	4
2.1.4. Vergleich der Kartendarstellungen	6
2.2. Messdatenintegration	8
2.2.1. Bestimmung der beobachteten Zellen	9
2.2.2. Bayesscher Filter	12
2.2.3. Zählverfahren	14
2.3. Vergleich verschiedener Systeme	17
2.4. Zusammenfassung	18
3. Konzept	19
3.1. Überblick	19
3.2. Bestimmen der Position	20
3.3. Bestimmen der beobachteten Zellen	22
3.4. Integrationsverfahren	23
3.5. Mapdatenbank	24
4. Implementierung	26
4.1. Überblick	26
4.2. Hardwaredesign	26

4.3. Softwaredesign	27
4.3.1. Mapping Software am Quadrocopter	27
4.3.2. SD Karte	29
4.3.3. Mapping Software am PC	31
4.3.4. GUI am PC	34
5. Evaluierung	38
5.1. Überblick	38
5.2. Optimierung	38
5.2.1. Reichweite des Sensors	39
5.2.2. Sensoröffnungswinkel	40
5.2.3. Zellengröße	41
5.2.4. Bayes Wahrscheinlichkeiten	42
5.2.5. Zählvariablen	44
5.3. Vergleich zwischen Bayes- und Zählverfahren	45
5.4. Auswirkungen der Wahrscheinlichkeitsverteilung	47
5.4.1. Auswirkungen auf das Bayes-Verfahren	47
5.4.2. Auswirkungen auf das Zählverfahren	48
5.5. Zusammenfassung	48
6. Diskussion und Ausblick	50
6.1. Diskussion	50
6.2. Ausblick	51
6.2.1. Fehlerhaft bzw. schlecht erkannte Zellen	52
6.2.2. Leistungsprobleme	53
A. Anhang	56
A.1. Messdaten umrechnen	56
A.1.1. Messdaten in Quadrocoptersystem umwandeln	56
A.1.2. Umwandlung in Weltkoordinaten	56
A.2. Bresenham Sektoreinteilung	57

A.3. Optimierung der Mapping Parameter	58
A.3.1. Verwendete Parameter	58
A.3.2. Ergebnisse der Optimierung	60
A.4. Ergebnis des Vergleichs der Mapping-Verfahren	66
A.5. Aufbau einer Map	68
A.5.1. Bayes-Verfahren	68
A.5.2. Zählverfahren	69
A.6. Auswirkung der Wahrscheinlichkeitsverteilung	70
A.6.1. Verwendete Parameter	70
A.6.2. Bayes-Verfahren	71
A.6.3. Zählverfahren	72

1. Einleitung

In gefährlichen Umgebungen, die ein hohes Risiko für Menschen bergen, bieten sich autonome unbemannte Luftfahrzeuge an. Der Lehrstuhl Aerospace Information Technology der Universität Würzburg entwickelt ein solches System für den Indoor-Betrieb. Im AQopterI8-Projekt wird kostengünstige Hardware verwendet, die auch in schwierigen Situationen, wie etwa verräucherten Räumen, funktioniert. Gerade im Indoor-Betrieb kann sich die Umgebung, besonders durch äußere Einflüsse, stark verändern. Aber auch Möbelstücke können ein Hindernis darstellen. Für ein autonomes System ist es daher besonders wichtig, selbst in der Lage zu sein, seine Umgebung zu erfassen und daraus eine Map zu erstellen, beziehungsweise eine gegebene Map zu erweitern und nach dieser Map selbstständig zu navigieren. Mapping bezeichnet dabei den Vorgang aus der zur Verfügung stehenden Sensorik eine Karte der Umgebung zu erstellen. Über die Sensorik nimmt der Quadrocopter seine Umwelt wahr. Diese Wahrnehmung wird beim Mapping über geeignete Verfahren in eine Repräsentation seiner Umgebung, der Map, übersetzt. Unterschiedliche Sensoren besitzen hierbei besondere Charakteristiken, die in den gewählten Verfahren beachtet werden müssen. Im Gegensatz zu anderen Quadrocoptern nutzt der AQopterI8 Ultraschallsensoren, anstelle der häufig verwendeten Laser-Scanner. Es kann mit wenigen Ultraschallsensoren die gesamte Umgebung in 360° erfasst werden. Dies stellt jedoch neue Herausforderungen an das Mapping, da mit Ultraschallsensoren nicht die exakte Position des erfassten Hindernisses bekannt ist. Auch kann das entstehende Mapping weiter zu einer SLAM-Lösung (Simultaneous Localization and Mapping) erweitert werden. Der Einsatz kostengünstiger Hardware und die damit verbundenen Besonderheiten machen dieses System besonders interessant. Die Arbeit stellt dabei gängige Verfahren zum Mapping vor. Diese werden an die besonderen Gegebenheiten wie die Sensorik angepasst und zu einem Konzept vereint. Im darauf folgendem Kapitel wird auf die Implementation der vorgestellten Mapping-Lösung und die damit entstehenden Probleme eingegangen. Das System wurde im Anschluss evaluiert und die Ergebnisse diskutiert.

2. Stand der Technik

Der Stand der Technik ermöglicht es, auch direkt auf kleinen Robotern Karten der Umgebung zu erstellen. Mapping beschreibt die Erfassung der Umwelt, wie sie von einem Roboter wahrgenommen wird. Dabei werden die von der Onboard Sensorik erfassten Messwerte gesammelt und zu einer möglichst umfassenden und aussagekräftigen Beschreibung der Umwelt zusammengefasst. Diese Beschreibung wird Map beziehungsweise Karte genannt. Die Sensorwerte werden ermittelt, nach Bedarf umgerechnet und inkrementell in bereits bestehende Karten integriert. Dadurch ist es möglich einen Roboter bereits im vorraus mit einer Karte auszustatten, die ergänzt werden soll, oder den Roboter eine neue Karte der Umgebung erstellen zu lassen. Ein dabei häufig auftretendes Problem ist unter anderem der Speichermangel der verwendeten Systeme. Je größer das Gebiet in Relation zur Sensorreichweite ist, desto schwieriger wird die Erstellung einer akkuraten Karte. Sensorrauschen ist ein weiteres Problem, welches die Erstellung einer Karte erschwert. Gerade in Indoor-Anwendungen, in denen sich verschieden Orte stark ähneln können, ist es schwer zwischen diesen zu unterscheiden. Auch zyklische Umgebungen, wie sie häufig in Gebäuden anzutreffen sind, stellen ein großes Problem dar, wenn keine globale Lokalisation wie im Outdoor-Bereich möglich ist. Da zur Erstellung einer Karte eine Position notwendig ist und diese Position wiederum aus der Karte und der aktuellen Messung berechnet werden soll, wird zur Lösung dieses Problems häufig SLAM (Simultaneous Localization and Mapping) Verfahren eingesetzt [Thrun et al., 2005]. Dabei wird aus der aktuellen Messung die wahrscheinlichste Position in der Map berechnet, die zur Navigation und weiteren Verfeinerung und Ergänzung der bestehenden Karte notwendig ist. Fehler der Odometrie und Inertialsensoren können so minimiert werden. Zur Darstellung der Karte existieren bereits erprobte Verfahren von denen die am häufigsten verwendeten in den nachfolgenden Kapiteln erläutert und die Vor- und Nachteile verglichen werden.

2.1. Mapdarstellung

2.1.1. Geometrische Darstellung

Die geometrische Darstellung nutzt einfache geometrische Objekte (Features) wie zum Beispiel Punkte, Linien und Kurven um die Karte darzustellen. Die Karte besteht dabei aus im voraus bereits festgelegten Features. Diese Features werden aus den Messungen extrahiert, um damit bestmöglich die Umgebung abzubilden. Dabei wird mit steigender Komplexität der geometrischen Objekte die Komplexität der Karte und der Detailgrad, wie in Abbildung 2.1 dargestellt, reduziert. Da die Erkennung der Features online erfolgt, eignen sich nur simple Verfahren zur Feature-Erkennung. Einfache Verfahren liefern in künstlichen Umgebungen wie zum Beispiel Bürogebäuden meist gute Ergebnisse und funktionieren solange das zu kartografierende Gebiet nicht zu komplex ist. Da die zu kartografierende Umgebung jedoch nicht bekannt ist, können die gewählten geometrischen Objekte diese unter Umständen nicht ausreichend darstellen und es kann zu Fehlerkennungen kommen. Fehlerkennungen können zudem durch schlechte Sensorqualität entstehen. In dieser Form der Darstellung kann zudem keine Aussage über den Raum zwischen den erfassten Objekten gemacht werden. Weiterhin ist die geometrische Map beschränkt in der Darstellung von Wahrscheinlichkeiten, welche für SLAM (Simultaneous Localization and Mapping) Verfahren benötigt werden. [Haehnel, 2004]

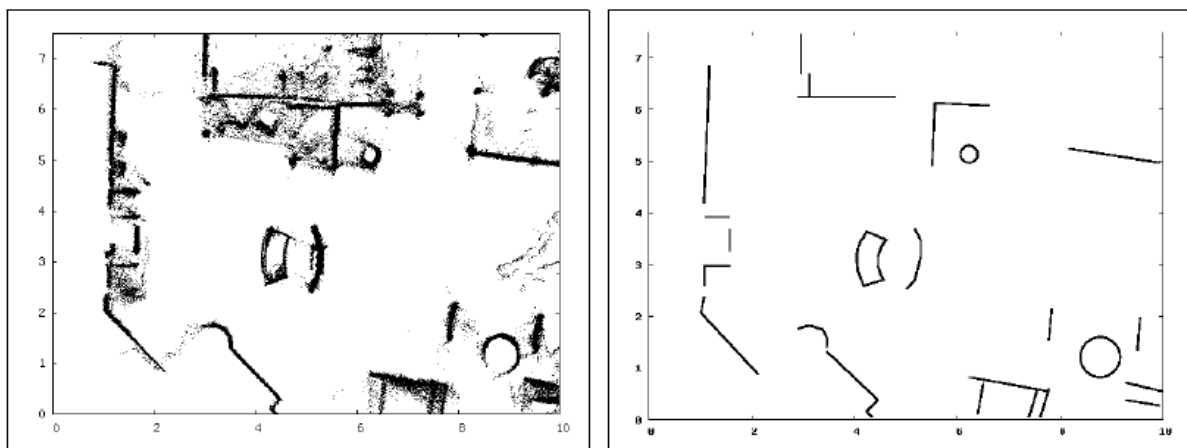


Abbildung 2.1.: Geometrische Darstellung mit Punkten (links) und Linien und Kurvensegmenten (rechts). [Haehnel, 2004]

2.1.2. Topologische Darstellung

Topologische Maps basieren auf Graphen. Hierbei wird die Bewegung des Roboters zwischen zwei Punkten als Kante betrachtet. An den Positionen des Quadrocopters werden Features aus den Sensordaten extrahiert, welche zusätzlich zu den Positionen als Knoten in den Graphen eingehen. Die Bewegung des Roboters zwischen den Positionen wird dabei als Kante in den Graphen integriert. Wird nach einer Bewegung ein bereits erkanntes Feature an der neuen Position erneut beobachtet, so wird eine Kante zwischen dem aktuellem Knoten und dem bereits bekannten Feature erstellt (Abbildung 2.2). [Thrun et al., 2005]

Da topologische Maps die Umgebung nicht komplett abbilden, sondern lediglich abstrahierte Features und die notwendige Bewegung zwischen diesen, sind sie effizienter als die geometrische und Grid Darstellung (vgl. Kapitel 2.1.3). Zudem wird keine exakte Position des Roboters benötigt, da die Bewegung zwischen Knoten relativ gespeichert wird und sich somit keine absoluten Fehler akkumulieren. Nachteile bestehen jedoch in der Extraktion und Wiedererkennung von Features, welche abhängig von der Genauigkeit und Eindeutigkeit der verwendeten Sensoren sind. [Thrun, 1998] [Haehnel, 2004]

2.1.3. Grid Darstellung

Die Grid Darstellung, auch als Evidence Grid [Haehnel, 2004] oder Occupancy Grid [Thrun et al., 2005] bezeichnet, unterteilt das zu kartografierende Gebiet in gleich große Zellen. Dabei ist die maximale Größe der Map durch die im Vorfeld festgelegte Auflösung und Zellengröße begrenzt. Mit sinkender Zellengröße steigt die Auflösung der Map, was jedoch einen steigenden Speicherverbrauch und eine höhere Rechenlast zur Folge hat. Weiterhin ist aufgrund beschränkter Rechenkapazität die Darstellung großer Gebiete mit hoher Auflösung nur bedingt möglich. Wie Abbildung 2.3 dargestellt, wird jede Zelle mit einem Wahrscheinlichkeitswert belegt, der Aufschluss darüber gibt, ob eine Zelle frei oder blockiert ist. Die Grid Darstellung eignet sich besonders zur Pfadplanung und zur Hindernisvermeidung, da im Gegenzug zur geometrischen oder topologischen Map leicht zwischen freien und belegten Bereichen unterschieden werden kann. Aufgrund der großen Datenmenge kann jedoch die Ausführung besonders rechenintensiv sein. Häufig werden für die Grid-Darstellung Quad-Trees verwendet. Hierbei wird jede Zelle rekursiv in vier wiederum gleichgroße Zellen aufgeteilt. Je tiefer die Kindknoten im Baum liegen, desto

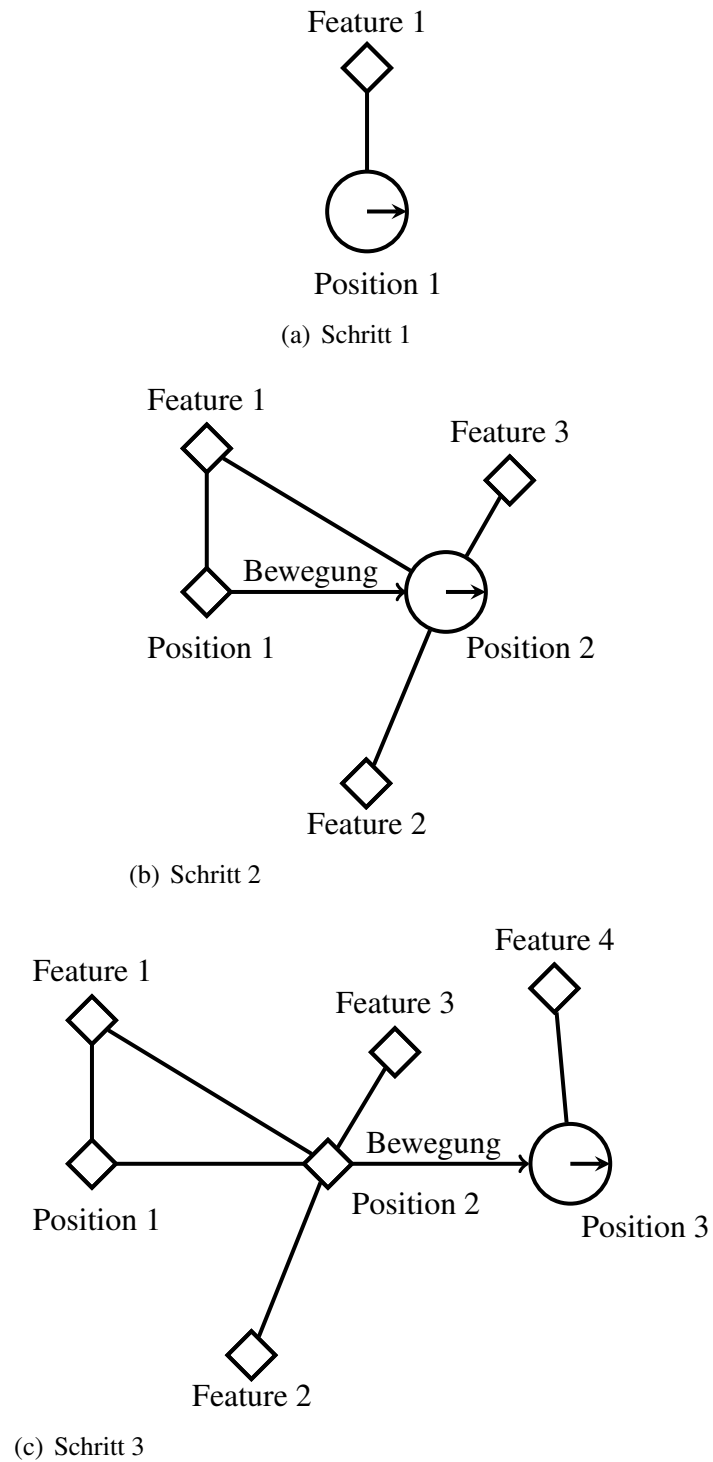


Abbildung 2.2.: Graph Aufbau zur topologischen Darstellung [Thrun et al., 2005]

höher ist die Auflösung. Durch den Quad-Tree wird zudem die Auflösung nur lokal und nicht global erhöht, was zu geringerem Speicherverbrauch und Rechenleistung gegenüber einer fixen Auflösung führt. [Hachnel, 2004]



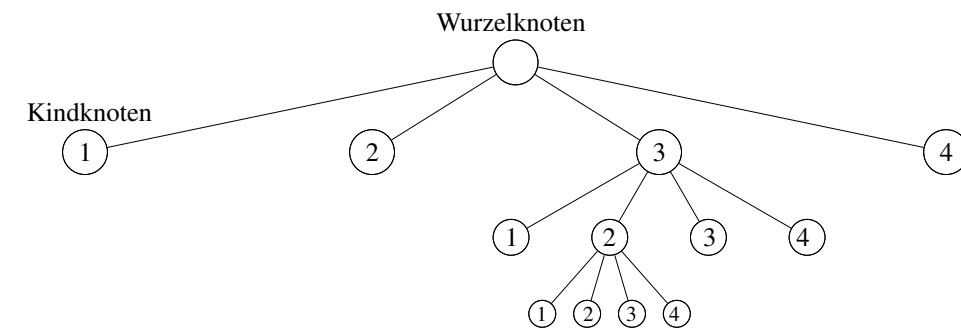
Abbildung 2.3.: Grid Map mit Wahrscheinlichkeiten. Weiße Bereiche sind frei, schwarze blockiert und graue undefiniert. [Hachnel, 2004]

2.1.4. Vergleich der Kartendarstellungen

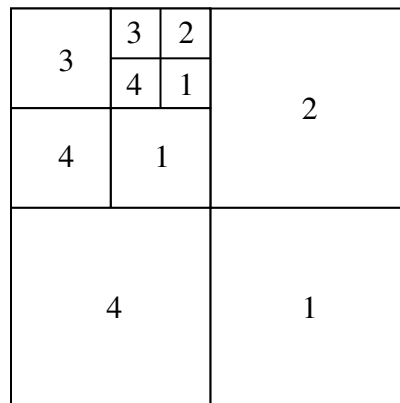
Alle vorgenannten Methoden zur Darstellung der Maps weisen deutliche Vor- und Nachteile auf. Zwar eignet sich die geometrische Darstellung für die Indoor-Anwendung, die mangelnde Information über die Zwischenräume und die damit verbundenen Probleme bei der Pfadplanung sowie der Hindernisvermeidung, welche für autonome Systeme ausschlaggebend sind, relativieren diesen Vorteil. Topologische und Grid Darstellung eignen sich deutlich besser zur Wegfindung durch Algorithmen wie Breiten-, Tiefen- und Dijkstra-Suche [Cormen et al., 2009]. Aufgrund der kompakten Speicherung von topologischen Maps sind diese besonders für größere Umgebungen geeignet. Ihr größtes Problem besteht in der Wiedererkennung von bereits bekannten Features. Die Grid Darstellung ist nur bedingt für größere Bereiche geeignet. Ein Vorteil ist die einfache Pfadplanung mit der leichten Erweiterbarkeit hin zu autonomen Systemen durch die einfache Integration von Wahrscheinlichkeiten. Der hohe Rechenaufwand und Speicherverbrauch kann durch geeignete Methoden wie Quad-Trees reduziert werden. Die Darstellung der Karte als Occupancy Grid eignet sich somit gut zur Repräsentation der Umwelt des Quadrocopters.

	Vorteile	Nachteile
Geometrisch	<ul style="list-style-type: none"> • Gut für künstliche Umgebungen 	<ul style="list-style-type: none"> • Keine Aussage über den Zwischenraum möglich • Wahl der geometrischen Features schwierig • Sensorqualität beeinträchtigt Featureerkennung
Topologisch	<ul style="list-style-type: none"> • Speichersparend • Keine absoluten Fehler 	<ul style="list-style-type: none"> • Keine genaue Abbildung der Umgebung • Wiedererkennung bereits erkannter Features schwierig
Grid	<ul style="list-style-type: none"> • Eignet sich gut zur Pfadplanung und Hindernisvermeidung • Detailgrad kann lokal erhöht werden 	<ul style="list-style-type: none"> • Rechen- und Speicherintensiv • Nicht für große Bereiche geeignet

Tabelle 2.1.: Vor- und Nachteil der einzelnen Mapping Verfahren.



(a) Quad-Tree



(b) Zellunterteilung durch den Quad-Tree

Abbildung 2.4.: Quad-Tree Darstellungen

2.2. Messdatenintegration

Nach dem Erfassen der Messwerte müssen diese in die Karte integriert werden. Dabei werden im Fall der Occupancy Grid Maps (vgl. Kapitel 2.1.3) alle vom Messwert erfassten Zellen bestimmt. Entsprechend dem Verfahren wird die Wahrscheinlichkeit ob die Zelle blockiert ist angepasst [Thrun et al., 2005]. Gängige Verfahren sind bayesscher Filter, Dempster-Shafer, Fuzzylogik und Zählverfahren [Haehnel, 2004]. Das Dempster-Shafer Verfahren erweitert die Wahrscheinlichkeit zur *Evidenz*, dass eine Zelle blockiert ist. Dazu wird die Aussagesicherheit der Sensoren und die Plausibilität des Ereignisses, ein Wahrscheinlichkeitsbereich mit unterer und oberer Grenze, zusammengefasst. Dadurch können unsichere Aussagen aus mehreren Sensoren zu einer Evidenz mit höherer Aussagesicherheit kombiniert werden. Die Fuzzylogik berechnet nicht die Wahrscheinlichkeit sondern die Unschärfe eines Systems. Dabei werden die Sensorwerte durch geeignete Funktionen auf die Fuzzy-Mengen, Mengen in denen Elemente auch nur teilweise enthalten sein können, abgebildet (Fuzzyfizierung). Die Fuzzy-Logik berechnet dann anhand im voraus

festgelegter Regeln die Unschärfe des Systems [wiki, 2012]. Das häufig verwendete Bayessche Aktualisierungsverfahren und das Zählverfahren werden in den nächsten Kapiteln, zusammen mit einem Algorithmus zur Bestimmung der von der Messung erfassten Zellen, vorgestellt.

2.2.1. Bestimmung der beobachteten Zellen

Da der Quadrocopter bzw. Roboter sich in der Umgebung bewegt, ist es wichtig eine möglichst aussagekräftige Map zu erstellen, die nicht nur die blockierten Zellen enthält, sondern ebenfalls eine Aussage über freie Zellen zulässt. Zur Bestimmung der vom Sensor beobachteten Zellen stehen zwei Verfahren zur Auswahl. Einerseits ein Line-Following Algorithmus ([Klawonn, 2012]) und andererseits ein sogenanntes Sektorenmodell, das den Öffnungswinkel der Ultraschallsensoren berücksichtigt, wie in [Thrun et al., 2005] beschrieben.

Line-Following Algorithmus (Bresenham) Dieser Algorithmus bestimmt die Zellen die zwischen dem Ausgangs- und Endpunkt der Messung liegen. Da beide Punkte der Messung durch die Quadrocopterposition und den Messwert vorgegeben sind, kann hierzu der einfache Line-Following Algorithmus von Bresenham genutzt werden. Dabei wird anhand einer Entscheidungsvariable D entschieden ob der Messstrahl näher an der Zelle E oder NE liegt (vgl. Abbildung 2.5). Die Entscheidungsvariable wird angepasst und der Algorithmus von der neuen Zelle aus wiederholt, bis die Zielzelle erreicht ist. Dieses Verfahren ist schnell, reduziert jedoch Ultraschallsensoren mit einem Öffnungswinkel auf einen Strahl. Aus diesem Grund ist es für die verwendeten Ultraschallsensoren ungenauer als das Sektorenmodell, benötigt allerdings weniger Rechenleistung.

Der Algorithmus beruht darauf, dass es nur zwei Zellen, E und NE, gibt, die nach einer bereits vom Strahl erfassten Zelle (x_p, y_p) liegen können. Die Steigung des Strahls muss im Intervall $[0, 1]$ liegen. Sollte die Steigung größer als 1 sein, so müssen die x und y Werte vertauscht werden um die Steigung im Intervall $[0, 1]$ zu halten. Zuerst wird die Differenz $dx = x_m - x_q$ und $dy = y_m - y_q$ aus der Position (x_q, y_q) und der Messung (x_m, y_m) gebildet. Danach wird die Entscheidungsvariable D nach Gleichung 2.1 initialisiert und mit jedem Schritt nach 2.2 aktualisiert. Bei jedem Durchlauf

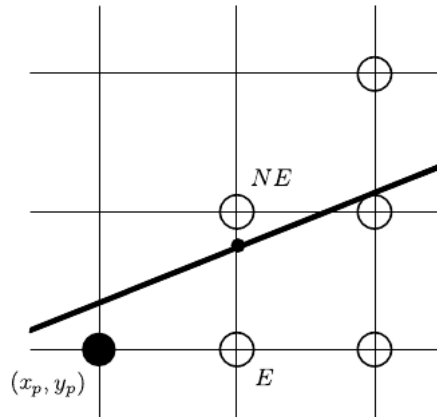


Abbildung 2.5.: Die zwei möglichen Zellen E und NE des Bresenham Algorithmus. [Klawonn, 2012]

des Algorithmus wird die x -Koordinate der Zelle zusätzlich um 1 inkrementiert. Die Herleitung der Gleichungen 2.1 und 2.2 kann in [Klawonn, 2012] nachgeschlagen werden.

$$D_{init} = 2 \cdot dy - dx \quad (2.1)$$

$$D_{new} = D_{old} + \Delta \text{ mit } \begin{cases} 2 \cdot dy & \text{für } D_{old} \leq 0 \\ 2 \cdot (dy - dx) & \text{für } D_{old} > 0 \end{cases} \quad (2.2)$$

Die beiden möglichen Werte für Δ werden zu Beginn des Algorithmus einmalig berechnet und zwischengespeichert. Nach jedem Schritt wird anhand D entschieden, ob die Zelle NE oder E vom Strahl durchlaufen wird. Hierbei gilt, dass die Zelle E gewählt wird für $D \leq 0$, und NE für $D > 0$. Wurde die Zelle NE gewählt, muss in diesem Schritt die y -Koordinate der Zelle noch um 1 inkrementiert werden. Abbruchbedingung ist das Erreichen der vom Sensor erfassten Zielzelle.[Klawonn, 2012]

Sektorenmodell Das Sektorenmodell bestimmt alle Zellen, die innerhalb eines Kreissektors liegen. Dabei wird auf den einfachen, in [Thrun et al., 2005] beschriebenen Algorithmus zurückgegriffen.

Der Algorithmus 2.1 berechnet den Abstand der aktuellen Zelle zum Sensor r , wobei x und y die Sensorkoordinaten sowie x_i und y_i der Schwerpunkt bzw. der Mittelpunkt der zu prüfenden Zelle m_i sind. In Zeile 4 wird der Winkel zwischen der Sensorausrichtung θ und der Zelle berechnet. In Zeile 6-11 wird entschieden, ob sich die Zelle innerhalb des Sensorsektors befindet. Wenn die

Zelle sich nicht im Sektor befindet (Zeile 6 und 7), erfolgt keine Integration in die aktuelle Karte. α stellt hierbei die Zellengröße und β den Öffnungswinkel des Sensors dar. Liegt die Zelle mit mindestens $\alpha/2$ auf dem zugehörigen Kreisbogen (Zeile 8 und 9) und der gemessene Wert z_t ist kleiner als die Sensorreichweite z_{max} , wird die Zelle als blockiert angesehen. Bei Zellen die sich im Kreissektor (Zeile 10 und 11) befinden, wird die Zelle als frei angenommen. Das Sektorenmodell ist aufwendiger in der Ausführung als der Bresenham-Algorithmus, liefert allerdings genauere Ergebnisse, da es ein realistischeres Modell der Ultraschallsensoren darstellt. Die Anwendung des Sektorenmodells ist in Abbildung 2.6 dargestellt.

```

1 inverse_sensor_model( $m_i$ ,  $x_t$ ,  $z_t$ )
2    $x_i$ ,  $y_i$  sei der Schwerpunkt der Zelle  $m_i$ 
3    $r = \sqrt{(x_i - x)^2 + (y_i - y)^2}$ ;
4    $\phi = \mathbf{atan2}(y_i - y, x_i - x) - \theta$ ;
5
6   if ( $r > \min(z_{max}, z_t + \alpha/2)$  or  $|\phi| > \beta/2$ )
7     return;
8   if ( $z_t < z_{max}$  and  $|r - z_{max}| < \alpha/2$ )
9     return blocked;
10  if ( $r \leq z_t$ )
11    return free;

```

Algorithmus 2.1: Sektorenmodell [Thrun et al., 2005]

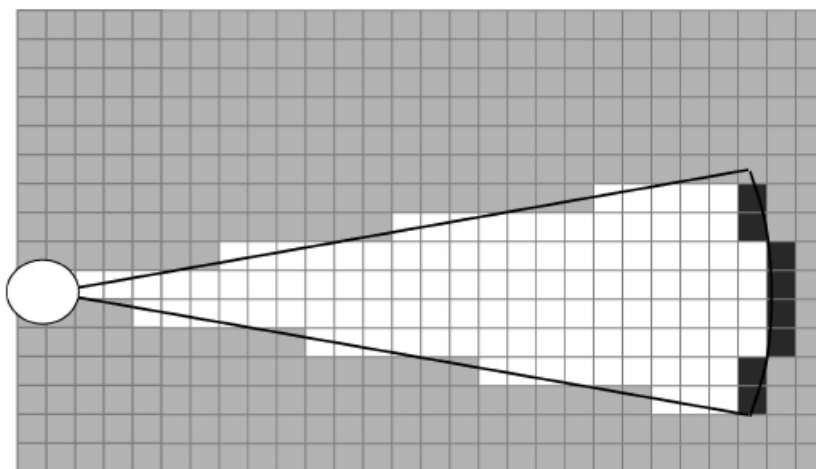


Abbildung 2.6.: Sektormodell für eine Messung. Je dunkler eine Zelle ist, desto wahrscheinlicher ist sie blockiert. [Thrun et al., 2005]

2.2.2. Bayesscher Filter

Die Bayes-Methode berechnet durch Anwendung des Bayestheorem die Wahrscheinlichkeit einer Zelle. Dabei verwendet dieses Verfahren einen binären bayesschen Filter, der darauf beruht, dass sich der Zustand der Umgebung nicht ändert, und aus den zwei Zuständen blockiert x ($p(x)$) und nicht blockiert $\neg x$ ($1 - p(x)$) besteht. Diese beiden Zustände werden über den bayesschen Filter auf die Map abgebildet. Durch diese binäre Betrachtungsweise wird der Vertrauenswert (belief) zu einer Funktion aller durchgeführten Messungen $z_{1:t} = z_1, \dots, z_t$ bis zum Zeitpunkt t , mit $bel_t(x) = p(x|z_{1:t})$. Analog zu $z_{1:t}$, stellt $x_{1:t}$ alle Positionen des Quadrocopters bis t dar. Es wird ein inverses Sensormodell $p(x|z_t)$ anstelle des normalen Sensormodells $p(z_t|x)$ verwendet. Das inverse Modell stellt eine Verteilung über die Zustände als Funktion der Messung z_t zum Zeitpunkt t dar. Der Filter wird, wie in 2.4 dargestellt, aus dem Bayestheorem (vgl. Gl. 2.3) hergeleitet.

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)} \text{ mit } p(B) > 0 \quad (2.3)$$

Wendet man das Bayestheorem auf bel_t an, so erhält man die Gleichung 2.4. Dasselbe wird für das Sensormodell (Gl. 2.5) durchgeführt.

$$\begin{aligned} p(x|z_{1:t}) &= \frac{p(z_t|x, z_{1:t-1}) p(x|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\ &= \frac{p(z_t|x) p(x|z_{1:t-1})}{p(z_t|z_{1:t-1})} \end{aligned} \quad (2.4)$$

$$p(z_t|x) = \frac{p(x|z_t) p(z_t)}{p(x)} \quad (2.5)$$

Gleichung 2.4 wird in 2.5 eingesetzt. Dadurch erhält man für den Zustand x die Gleichung 2.6. Analog dazu wird die Gleichung 2.7 hergeleitet.

$$p(x|z_{1:t}) = \frac{p(x|z_t) p(z_t) p(x|z_{1:t-1})}{p(x) p(z_t|z_{1:t-1})} \quad (2.6)$$

$$p(\neg x|z_{1:t}) = \frac{p(\neg x|z_t) p(z_t) p(\neg x|z_{1:t-1})}{p(\neg x) p(z_t|z_{1:t-1})} \quad (2.7)$$

Die bedingte Wahrscheinlichkeit $p(x|z_{1:t})$ (Gl. 2.6) wird durch $p(\neg x|z_{1:t})$ (Gl. 2.7) geteilt.

$$\begin{aligned} \frac{p(x|z_{1:t-1})}{p(\neg x|z_{1:t-1})} &= \frac{p(x|z_t)}{p(\neg x|z_t)} \cdot \frac{p(x|z_{1:t-1})}{p(\neg x|z_{1:t-1})} \cdot \frac{p(\neg x)}{p(x)} \\ &= \frac{p(x|z_t)}{1 - p(x|z_t)} \cdot \frac{p(x|z_{1:t-1})}{1 - p(x|z_{1:t-1})} \cdot \frac{1 - p(x)}{p(x)} \end{aligned} \quad (2.8)$$

Die Gleichung 2.8 lässt sich durch das *logarithmische Quotientenverhältnis (log odds ratio)*, wie in 2.9 definiert, und Gleichung 2.10 weiter vereinfachen, um den binären bayesschen Filter in 2.11 zu erhalten.

$$l(x) := \log \frac{p(x)}{1 - p(x)} \quad (2.9)$$

$$l_{t-1,i} = \log \frac{p(x|z_{1:t-1})}{1 - p(x|z_{1:t-1})} \quad (2.10)$$

$$\begin{aligned} l_t(x) &= \log \frac{p(x|z_t)}{1 - p(x|z_t)} + \log \frac{p(x|z_{1:t-1})}{1 - p(x|z_{1:t-1})} + \log \frac{1 - p(x)}{p(x)} \\ &= \log \frac{p(x|z_t)}{1 - p(x|z_t)} - \log \frac{p(x)}{1 - p(x)} + l_{t-1}(x) \end{aligned} \quad (2.11)$$

Da jedoch die Wahrscheinlichkeit aus 2.12 berechnet werden soll, muss der Filter angepasst werden. Dabei ist m die Karte als Occupancy Grid Map, die wie in Kapitel 2.1.3 bereits erwähnt aus endlich vielen Zellen $m = \{\mathbf{m}_i\}$ besteht. $p(\mathbf{m}_i)$ ist die Wahrscheinlichkeit, dass eine Zelle in der Karte blockiert ist. Wendet man dies auf den binären bayesschen Filter an, so ergibt sich Gleichung 2.13, die für jede vom Sensor beobachtete Zelle \mathbf{m}_i angewandt wird.

$$p(m|z_{1:t}, x_{1:t}) \quad (2.12)$$

$$l_{t,i} = \log \frac{p(\mathbf{m}_i|z_t, x_t)}{1 - p(\mathbf{m}_i|z_t, x_t)} - \log \frac{p(\mathbf{m}_i)}{1 - p(\mathbf{m}_i)} + l_{t-1,i} \quad (2.13)$$

$$l_t = \log \frac{p(\mathbf{m}_i|z_t, x_t)}{1 - p(\mathbf{m}_i|z_t, x_t)} \quad (2.14)$$

$$l_0 = \log \frac{p(\mathbf{m}_i)}{1 - p(\mathbf{m}_i)} \quad (2.15)$$

Nutzt man die Gleichungen 2.14 und 2.15, kann der Bayessche Filter verkürzt dargestellt werden:

$$l_{t,i} = l_t - l_0 + l_{t-1,i} \quad (2.16)$$

Der Term l_t ist hierbei das inverse Sensormodell. Der Term l_0 ist die *a priori* Wahrscheinlichkeit zu Beginn, bevor Messungen durchgeführt wurden. $l_{t,i}$ ist der in der Karte hinterlegte Wahrscheinlichkeitswert als logarithmisches Quotientenverhältnis. Der neue Wahrscheinlichkeitswert (vgl. Gl. 2.12) kann nach Gleichung 2.17 aus dem Quotientenverhältnis zurück gewonnen werden.

$$p(\mathbf{m}_i | z_{1:t}, x_{1:t}) = 1 - \frac{1}{1 + \exp(l_{t,i})} \quad (2.17)$$

Die Wahrscheinlichkeit $p(\mathbf{m}_i | z_t, x_t)$ wird wie folgt festgelegt (vgl. Gl. 2.18). Die aus [Haehnel, 2004] entnommenen Werte eignen sich gut für dieses Verfahren. Wahrscheinlichkeiten nahe den Extremwerten 0 und 1 eignen sich nicht, da die Messung sonst einen zu großen Einfluss gewinnt und l_0 und $l_{t-1,i}$ kaum in den Filter eingehen. Dies hat zur Folge, dass die Wahrscheinlichkeit der einzelnen Zellen stark schwanken.

$$p(\mathbf{m}_i | z_t, x_t) = \begin{cases} 0,2 & \text{für freie Zellen} \\ 0,8 & \text{für blockierte Zellen} \end{cases} \quad (2.18)$$

Freie Zellen sind wie bereits erwähnt Zellen, die vom Strahl des Sensors durchlaufen wurden bzw. die im Sensorsektor liegen. Als blockierte Zellen gelten solche, in denen der Strahl endet bzw. auf dem Kreisbogensegment des Sensorsektors liegen. [Thrun et al., 2005] [Haehnel, 2004]

2.2.3. Zählverfahren

Das Zählverfahren bietet neben dem bayesschen Filter eine einfache Möglichkeit zur Integration der Messwerte in eine Karte. Dafür werden zwei Zähler verwendet. Einer für die Anzahl der Beobachtungen der Zelle als frei (c_{free}) und einer für die Beobachtungen als blockiert (c_{occ}). Nach jeder Messung müssen lediglich die Zähler entsprechend erhöht werden. Die Herleitung des Zählverfahrens und der Zellwahrscheinlichkeit $p(z_{t,n} | x_t, m)$, mit der Messung $z_{t,n}$ zum Zeitpunkt t , der Karte m und der Position x_t , wird in Gleichung 2.19 beschrieben. Die Indikatorvariable $\zeta_{t,n}$ nimmt dabei den Wert 1 an, wenn der Sensor die maximale Reichweite zurück liefert. Wird nicht die maximale Reichweite gemessen, nimmt $\zeta_{t,n}$ den Wert 0 an. Dabei gibt die Funktion $f(x_t, n, k)$ den Index der Zelle von m zurück, die von der Messung erfasst wurde. Gibt der Sensor die maximale Reichweite zurück, werden alle Zellen, mit der Wahrscheinlichkeit $1 - m_{f(x_t, n, k)}$, als nicht blockiert

betrachtet. Analog zum bayesschen Filter werden bei Sensorwerten unter dem Maximum alle Zellen als frei betrachtet, die vom Strahl durchlaufen werden bzw. im Sensorsektor liegen und die Endzelle des Strahls, bzw. die Zellen auf dem Kreisbogenabschnitt, mit der Wahrscheinlichkeit $m_{f(x_t, n, z_{t, n})}$, werden als blockiert betrachtet. Die einzelnen Wahrscheinlichkeiten werden über Produkte zu $p(z_{t, n}|x_t, m)$ zusammengefasst.

$$p(z_{t, n}|x_t, m) = \prod_{k=0}^{z_{t, n}-1} (1 - m_{f(x_t, n, k)})^{\zeta_{t, n}} \cdot \left[m_{f(x_t, n, z_{t, n})} \cdot \prod_{k=0}^{z_{t, n}-1} (1 - m_{f(x_t, n, k)}) \right]^{(1-\zeta_{t, n})} \quad (2.19)$$

Durch die Indikatorvariable $\zeta_{t, n}$ wird die Gleichung 2.19 in zwei Terme aufgeteilt:

$$p(z_{t, n}|x_t, m) = \begin{cases} \prod_{k=0}^{z_{t, n}-1} (1 - m_{f(x_t, n, k)}) & \text{für max. Reichweite} \\ m_{f(x_t, n, z_{t, n})} \cdot \prod_{k=0}^{z_{t, n}-1} (1 - m_{f(x_t, n, k)}) & \text{für } z_{t, n} < \text{max. Reichweite} \end{cases} \quad (2.20)$$

Um die wahrscheinlichste Map $m(x_{1:t}, z_{1:t})$ aus den Messungen ($p(z_{t, n}|x_t, m)$) zu berechnen, muss das Maximum (*argmax*) der Wahrscheinlichkeit aller Messungen $p(z|x, m)$ bis zum Zeitpunkt T bestimmt werden. Unter der Annahme, dass die Messungen der einzelnen Sensoren, mit $n \in [1, N]$, zu einem Zeitpunkt unabhängig voneinander sind, ergibt sich somit Gleichung 2.21.

$$p(z|x, m) = \prod_{t=1}^T \prod_{n=1}^N p(z_{t, n}|x_t, m) \quad (2.21)$$

Durch die logarithmische Wahrscheinlichkeit erhält man so die folgende Gleichung:

$$\begin{aligned} \ln p(z|x, m) &= \ln \prod_{t=1}^T \prod_{n=1}^N p(z_{t, n}|x_t, m) \\ &= \sum_{t=1}^T \sum_{n=1}^N \ln p(z_{t, n}|x_t, m) \end{aligned} \quad (2.22)$$

Setzt man nun den Term für $z_{t,n} < \max.$ Reichweite ($\zeta_{t,n} = 0$) aus 2.20 in 2.22 ein, so erhält man die Gleichung 2.23.

$$\begin{aligned} \ln p(z|x, m) &= \sum_{t=1}^T \sum_{n=1}^N \left[\ln \left(m_{f(x_t, n, z_{t,n})} \cdot \prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t, n, k)}) \right) \right] \\ &= \sum_{t=1}^T \sum_{n=1}^N \left[\ln(m_{f(x_t, n, z_{t,n})}) + \sum_{k=0}^{z_{t,n}-1} \ln(1 - m_{f(x_t, n, k)}) \right] \end{aligned} \quad (2.23)$$

Dadurch, dass die Positionen x_1, \dots, x_t bereits bekannt sind, kann die neue Map $m(x_{1:t}, z_{1:t})$ durch die geschlossene Form für $m(x_{1:t}, z_{1:t})$ berechnet werden. Um die wahrscheinlichste Map zu berechnen, muss wie bereits erwähnt, das Maximum berechnet werden. Um die Gesamtwahrscheinlichkeit $m(x_{1:t}, z_{1:t})$ zu maximieren, wird über alle Zellen j , mit $j \in [1, J]$, der Map m summiert. Dadurch wird $m_{f(x_t, n, k)}$ und $m_{f(x_t, n, z_{t,n})}$ zu m_j . Zudem wird die Variable $I(y)$ eingeführt. $I(y)$ ist 1 für den Fall, dass y wahr ist und ansonsten 0. Dabei gilt das y wahr ist, wenn $f(x_t, n, k)$ beziehungsweise $f(x_t, n, z_{t,n})$ gleich j ist. Ansonsten ist y falsch. Dadurch kann die folgende Gleichung hergeleitet werden:

$$\begin{aligned} m(x_{1:t}, z_{1:t}) &= \underset{m}{\operatorname{argmax}} \sum_{j=1}^J \sum_{t=1}^T \sum_{n=1}^N I(f(x_t, n, z_{t,n}) = j) \cdot \ln(m_j) \\ &\quad + \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \cdot \ln(1 - m_j) \end{aligned} \quad (2.24)$$

Definiert man nun α_j und β_j wie in Gleichung 2.25 und 2.26 gezeigt, kann die Gleichung 2.27 durch Substitution in 2.24 hergeleitet werden.

$$\alpha_j := \sum_{t=1}^T \sum_{n=1}^N I(f(x_t, n, z_{t,n}) = j) \quad (2.25)$$

$$\beta_j := \sum_{t=1}^T \sum_{n=1}^N \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \quad (2.26)$$

$$m(x_{1:t}, z_{1:t}) = \underset{m}{\operatorname{argmax}} \sum_{j=1}^J \alpha_j \ln(m_j) + \beta_j \ln(1 - m_j) \quad (2.27)$$

Um nun das Maximum $m(x_{1:t}, z_{1:t})$ zu erhalten wird davon ausgegangen das die einzelnen Zellen m_j voneinander unabhängig sind. Damit kann $m(x_{1:t}, z_{1:t})$ über die Maxima der einzelnen Zellen

der Karte berechnet werden. Damit die Wahrscheinlichkeit einer einzelnen Zelle m_j maximal wird, muss gelten, dass die 1. Ableitung gleich 0 ist, wie in Gleichung 2.28 dargestellt.

$$\frac{\delta m}{\delta m_j} = \frac{\alpha_j}{m_j} - \frac{\beta_j}{1 - m_j} = 0 \quad (2.28)$$

Durch umstellen nach m_j ergeben sich zwei Zählvariablen pro Zelle (vgl. Gl. 2.29).

$$m_j = \frac{\alpha_j}{\beta_j + \alpha_j} \quad (2.29)$$

Die Variable β_j steht dabei für die Anzahl an Messungen, in denen die Zelle als nicht blockiert beobachtet wurde. α_j steht für die Anzahl der Beobachtungen der Zelle als blockiert. Beschreibt man die Wahrscheinlichkeit einer Zelle mit $p(m_j)$, so erhält man Gleichung 2.30. Dabei entspricht c_{free} und c_{occ} den Variablen β_j beziehungsweise α_j . Die Variablen c_{free} und c_{occ} summieren über den gesamten Zeitraum bis zum aktuellen Zeitpunkt T auf.

$$p(m_j) = \frac{c_{occ}}{c_{free} + c_{occ}} \quad (2.30)$$

[Haehnel, 2004]

2.3. Vergleich verschiedener Systeme

In der Robotik existieren bereits viele Mapping-Lösungen. Diese beruhen jedoch meist auf mobilen, bodengebundenen Systemen und sind aus diesem Grund für Quadrocopter nur bedingt geeignet ([Haehnel, 2004]). Auf vielen Quadrocoptern oder anderen UAVs werden Laser-Scanner ([Krause, 2012]), aber auch Videokameras ([Bošnak und Blažič, 2012]) verwendet. Bekannte Systeme unterscheiden sich somit meist in der verwendeten Sensorik zur Erfassung der Umgebung vom AQopterI8 der Universität Würzburg.

Ultraschallsensoren haben meist einen großen Öffnungswinkel anstelle eines einfachen Strahls wie er bei den häufig verwendeten Laser-Scannern üblich ist. Dadurch kann die Ausrichtung zu einem erkannten Hindernis nur in einem großen Toleranzbereich, dem Öffnungswinkel des Ultraschallsensors, angegeben werden. Die Verwendung einer auf den Boden gerichteten Kamera ([Bošnak und Blažič, 2012]) hingegen, kann Hindernisse in der Ebene, in der sich der Quadro-

copter bewegt, nicht erfassen. Somit ist eine Aussage über blockierte Bereiche im Gegenzug zu Ultraschallsensoren kaum möglich.

2.4. Zusammenfassung

In diesem Kapitel wurden allgemeine Verfahren zum Mapping vorgestellt. Dabei müssen jedoch die besonderen Anforderungen, die durch die Ultraschallsensoren entstehen, noch berücksichtigt werden. Darunter fallen unter anderem ein großer Öffnungswinkel des Sensors, die Winkelabhängigkeit des Sensors gegenüber der Umgebung und das Messrauschen der Sensoren ([Müller, 2011]). Aufbauend auf den vorgestellten Verfahren wird im folgenden Kapitel eine neue Mapping-Lösung erarbeitet, die diese Besonderheiten berücksichtigt. Die Mapping-Lösung wird dabei an die Sensorik des AQuopterI8 angepasst.

3. Konzept

3.1. Überblick

Aus den bereits vorgestellten Verfahren werden in diesem Kapitel die passenden ausgewählt und zu einem, auf die besonderen Anforderungen zugeschnittenen, Konzept ausgearbeitet.

Den Ablauf des Mappingverfahrens zeigt Abbildung 3.1. Zuerst wird die Umgebung mit den Sensoren erfasst und die Position des Quadrocopters bestimmt. Danach wird, wie in A.1 beschrieben, die gewonnen Messdaten in das Weltkoordinatensystem gedreht, um eine Integration in die Map zu ermöglichen. Als Map wird hierbei eine Occupancy Grid Map verwendet. Diese eignet sich gut zur Darstellung der Umgebung in Gebäuden. Es bietet zudem den Vorteil einer einfachen Pfadplanung und Hindernisvermeidung. Die Grid Map bietet zudem die Möglichkeit verschiedener Arten der Messdatenintegration und Datenstrukturen in denen die Map intern gespeichert wird. Alle zwischen der vom Sensor gemessenen Zelle und der Position des Quadrocopters liegenden Zellen können als frei angenommen werden [Thrun et al., 2005]. Aus diesem Grund werden diese Zellen durch die in Abschnitt 2.2.1 vorgestellten Algorithmen bestimmt. Sobald alle Zellen bekannt sind, wird das Integrationsverfahren auf diese angewandt. Hier kommen das Zählverfahren und der bayessche Filter zum Einsatz. In der Karte werden hierfür die Zählvariablen, sowie die Wahrscheinlichkeit nach Bayes gespeichert. Dies ermöglicht es aus einer Map zwei verschiedene Repräsentation, nach dem Bayes- und Zählverfahren, zu erstellen und diese zu vergleichen. So kann das besser geeignete Verfahren bestimmt werden. Bisher noch nicht erfasste Zellen müssen sofern sie vom Sensor erfasst wurden in der Datenbank (vgl. Kapitel 3.5) angelegt werden. Bereits vorhandene Zellen können direkt verwendet werden. Zur Speicherung der Zellen wird ein Quad-Tree genutzt.

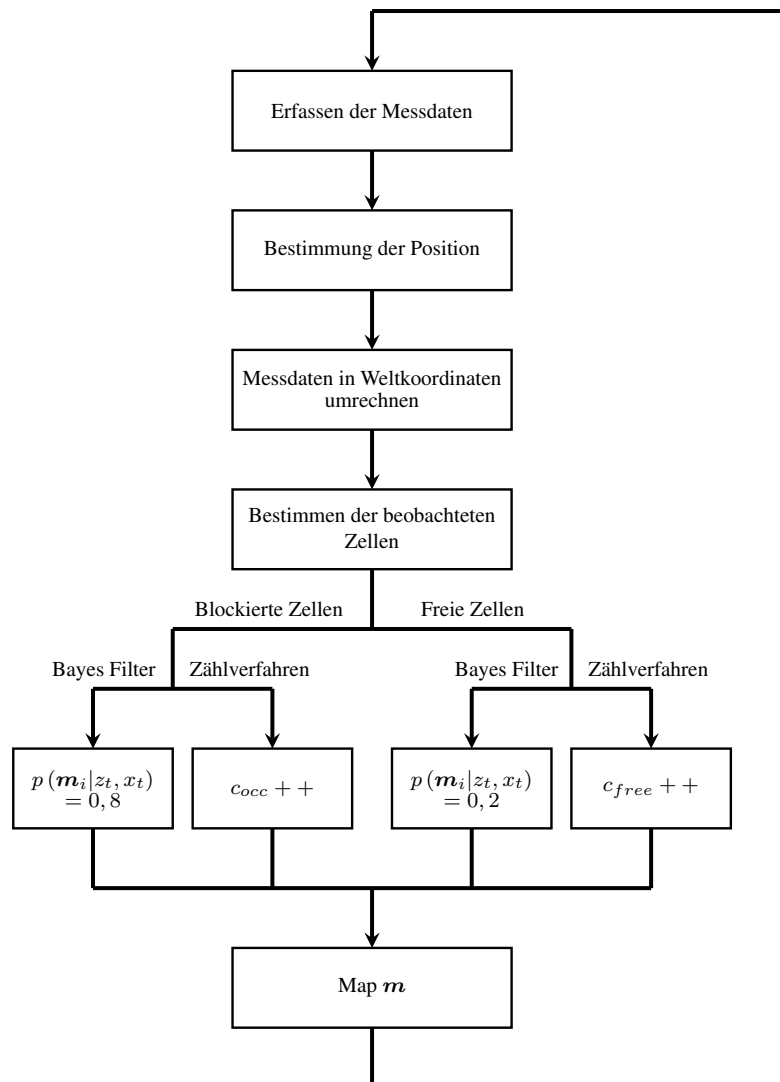


Abbildung 3.1.: Ablauf des Mapping Verfahrens.

3.2. Bestimmen der Position

Da das Mapping ohne bekannte Position nicht möglich ist, muss sie über die zur Verfügung stehenden Sensoren bestimmt werden. In diesem Kapitel wird ein Verfahren ausgearbeitet, um die Position aus den Sensorwerten zu berechnen. Zur Vereinfachung wird die Positionsbestimmung nur zweidimensional durchgeführt und davon ausgegangen, dass die Messungen nicht fehlerbehaftet sind. Es werden 4 Sensoren genutzt, von denen Sensor 1 und 3 parallel zur x -Achse, und die Sensoren 2 und 4 parallel zur y -Achse messen.

Zu Beginn wird die Position x_0 und Orientierung Ψ des Quadrocopters als Nullpunkt festgelegt. Da über Abstandssensoren nicht festgestellt werden kann, ob sich Ψ ändert, wird die Orientierung

als konstant angenommen. $u_{t,n}$ steht für eine Messung zum Zeitpunkt t des Sensors n . Die Abstandsmessungen in $u_{t,1}$ und $u_{t,3}$, zum Zeitpunkt t , werden zur Bestimmung der Position in x -Richtung verwendet. Analog dazu werden $u_{t,2}$ und $u_{t,4}$ für die y -Richtung genutzt. Die aktuelle Position $x_t = (x_{t,x}, x_{t,y})$ zum Zeitpunkt t berechnet sich aus der vorherigen Position x_{t-1} sowie der Differenz Δu der beiden Messungen u_t und u_{t-1} mit $\Delta u = u_t - u_{t-1}$, wie in Gleichung 3.1 dargestellt. Messungen die die maximale Reichweite haben, werden vernachlässigt, da sie den gemittelten Wert zu stark verfälschen würden.

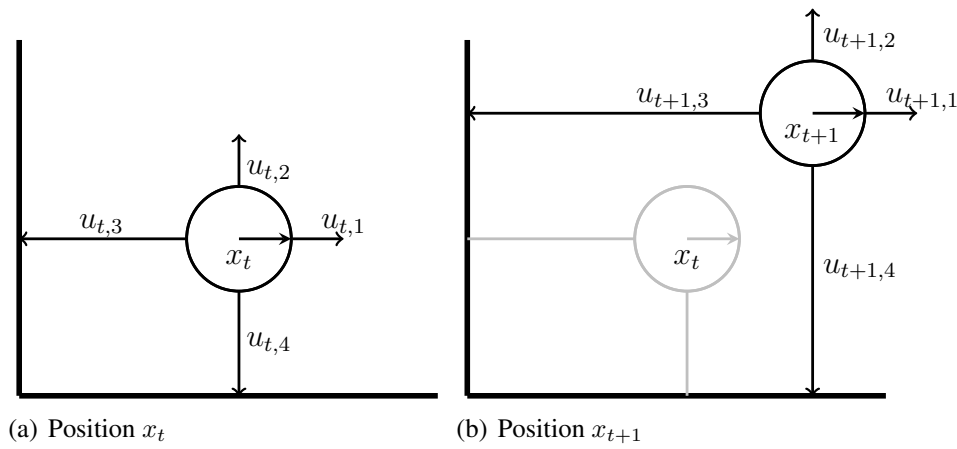


Abbildung 3.2.: Positionsbestimmung durch die Sensoren.

$$x_{t,x} = x_{t-1,x} + u_{t,1,3} - u_{t-1,1,3}$$

$$x_{t,y} = x_{t-1,y} + u_{t,2,4} - u_{t-1,2,4} \quad (3.1)$$

Die Positionsbestimmung durch Abstandsmessung alleine ist jedoch nur für den Idealfall geeignet. In großen Räumen, wenn alle Sensoren die maximale Reichweite zurück liefern, kann diese Methode nicht mehr angewandt werden. Selbes gilt ebenfalls für lange Flure, in denen eine Bewegung entlang des Flures nicht mehr nachvollzogen werden kann. In diesen Situationen kann der Quadrocopter nicht mehr ohne zusätzliche Sensoren entscheiden, ob ein Versatz in der Umgebung vorliegt oder er sich in Richtung eines Hindernisses bewegt hat.

Eine weitere Möglichkeit der Positionsbestimmung wäre unter andere die Nutzung des optischen Flusses oder die Verwendung einer PMD-Kamera (Photomischdetektor). Weitere Sensoren, wie zum Beispiel Inertialsensoren, können das Verfahren zudem weiter verbessern. Für die Evaluierung

des Systems steht zudem ein optisches Tracking System zur Verfügung, mit dem die Position des Quadrocopters während der Evaluierung bestimmt werden kann.

3.3. Bestimmen der beobachteten Zellen

Da der Bresenham Algorithmus auf einer Vereinfachung des Sensormodells zu einem Strahl beruht, eignet er sich für den Einsatz im Quadrocopter der über eine eingeschränkte Leistung gegenüber normalen Computern besitzt. Dabei handelt es sich jedoch um ein stark vereinfachtes Sensormodell welches Nachteile mit sich bringt. So werden durch den breiten Öffnungswinkel des Sensor Objekte, die seitlich des Sensors liegen, in das Zentrum des Sensors verschoben (vgl. Abbildung 3.3).

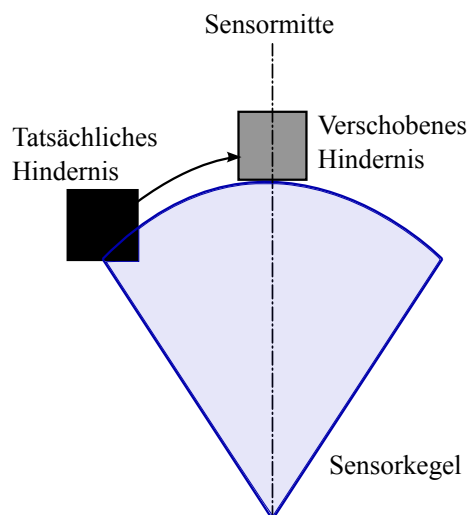


Abbildung 3.3.: Verschiebung eines Hindernisses durch das Strahlenmodell

Steht ausreichend Rechenleistung zur Verfügung, wird das Sektormodell `inverse_sensor_model()` (siehe Algorithmus 2.1) genutzt. Um das Modell noch zu verfeinern und die Winkelabhängigkeit der Messung zu beachten, wurde es um eine einfache lineare Wahrscheinlichkeitsverteilung erweitert. Über den Abstand r und dem Winkel zwischen der Sensorausrichtung und der aktuellen Zelle ϕ , wird der Abstand zum Mittelstrahl über $r \cdot \sin(\phi)$ berechnet. Das Verhältnis zwischen dem Abstand zur Sektormitte und dem maximalen Wert $z_t \cdot \sin(\varphi)$ dient als lineare Verteilung. Dabei ist φ der halbe Öffnungswinkel und z_t die gemessene Distanz.

Bayes Wahrscheinlichkeit Für den Bayes-Filter berechnet sich die Wahrscheinlichkeit für freie Zellen und blockierte Zellen nach Gleichung 3.2 bzw. 3.3. Es ergibt sich eine Wahrscheinlichkeitsverteilung für freie Zellen im Intervall $\left[p(\mathbf{m}_i|z_t, x_t)_{free}, \frac{1}{2}\right]$ und für blockierte Zellen im Intervall $\left[\frac{1}{2}, p(\mathbf{m}_i|z_t, x_t)_{occ}\right]$.

$$p_{free} = p(\mathbf{m}_i|z_t, x_t)_{free} + \left[\frac{r \cdot \sin(\phi)}{z_t \cdot \sin(\varphi)} \cdot \left(\frac{1}{2} - p(\mathbf{m}_i|z_t, x_t)_{free} \right) \right] \quad (3.2)$$

$$p_{occ} = p(\mathbf{m}_i|z_t, x_t)_{occ} - \left[\frac{r \cdot \sin(\phi)}{z_t \cdot \sin(\varphi)} \cdot \left(p(\mathbf{m}_i|z_t, x_t)_{occ} - \frac{1}{2} \right) \right] \quad (3.3)$$

Zählverfahren Für das Zählverfahren werden die Inkrementierungsvariablen angepasst. Anstelle einer einfachen Erhöhung des Wertes um 1, wird die Verteilung mit einer Konstanten m multipliziert. m bestimmt die Steigung der Verteilung. Je größer m ist, desto stärker werden Zellen in der Nähe des Mittelstrahls gewertet und die Inkrementierungsvariable wird entsprechend größer. Gleichung 3.4 kann sowohl für freie, als auch für blockierte Zellen als Verteilungsfunktion verwendet werden.

$$c_{inc} = \left(1 - \frac{r \cdot \sin(\phi)}{z_t \cdot \sin(\varphi)} \right) \cdot m \quad (3.4)$$

3.4. Integrationsverfahren

In Abschnitt 2.2.2 und 2.2.3 wurden zwei Möglichkeiten zur Integration der Messdaten in die bestehende Map vorgestellt. Während das Zählverfahren auf einfachem Inkrementieren zweier Zähler und dem Verhältnis der beiden zueinander aufbaut, wird bei der Bayes-Methode durch einen binären Filter die Wahrscheinlichkeit berechnet. Das Zählverfahren bietet den Vorteil einfach in der Implementierung und nicht rechenintensiv zu sein. Allerdings werden intern zwei Variablen für die Zähler benötigt, was den Speicherverbrauch gegenüber dem Bayes Filter erhöht. Der Bayes Filter hingegen konvergiert schneller zu den Extermwerten 0 oder 1 [Thrun et al., 2005]. Um einen Vergleich zwischen den beiden Verfahren am realen System und den damit verbundenen Besonderheiten der Ultraschallsensoren zu ermöglichen, werden beide Verfahren implementiert.

3.5. Mapdatenbank

Die erfassten Daten der Sensoren werden zu einer Map zusammengefasst. Hierzu ist es notwendig eine effiziente Speicherung zu haben. Dabei bietet sich für die Occupancy Grid Map entweder die Möglichkeit als Graph oder als Quad-Tree an, bei dem die Zellen immer weiter in vier kleinere Zellen aufgeteilt werden. Der Graph besteht hierbei aus Knoten und Kanten. Die Zellen der Map stellen dabei die Knoten mit den Koordinaten dar, während die Kanten die Verbindung zu den vier direkt benachbarten Zellen widerspiegeln. Dadurch ist ein Knoten immer mit vier Kanten verbunden. Im Randbereich begrenzt sich die Anzahl der Kanten auf drei, beziehungsweise an den Ecken auf zwei Kanten. Größter Unterschied der beiden Datenstrukturen besteht im Speicherverbrauch. Dabei schneidet der Quad-Tree schlechter ab, da hier zusätzlich zu den kleinsten Zellen noch die in der Baumhierarchie darüber liegenden Zellen hinzukommen. Je kleiner die minimale Zellengröße, desto tiefer wird der Quad-Tree und umso mehr Zellen werden benötigt (vgl. 3.1). Dies resultiert in einem höheren Speicherverbrauch gegenüber der Graphendarstellung wie Abbildung 3.4 zeigt. Der Quad-Tree bietet jedoch den Vorteil unterschiedliche Auflösungen zu nutzen.

	1 cm	5 cm	10 cm
Graph	1.000.000	40.000	10.000
Quad-Tree	1.398.100	87.380	21.844

Tabelle 3.1.: Zellenanzahl verschiedener Zellengrößen bei einer Fläche von 100m²

Wird davon ausgegangen, dass der Wahrscheinlichkeitswert der Occupancy Grid Map in einer 4byte großen Variable gespeichert wird, ergibt sich bei einer Zellengröße von 1cm und einer zu kartografierenden Fläche von 100m², ein Speicherbedarf von 3,8MB für die Graphenstruktur und 5,3MB für den Quad-Tree. Für kleinere Zellengrößen eignen sich beide Verfahren nur in räumlich stark begrenzten Umgebungen, da der Speicherbedarf quadratisch wächst. Dieser berechnet sich für den Quad-Tree aus der benötigten Baumtiefe mit $d = \log_4 \left(\frac{a}{c} \right)$ mit der zu kartografierenden Fläche a und der Zellenfläche c . Die gesamte Baumgröße für die Tiefe d berechnet sich mit $d = \sum_{i=1}^d 4^i$ und in geschlossener Form $\frac{4}{3} (4^d - 1)$. Aufgrund der unterschiedlichen Auflösungen

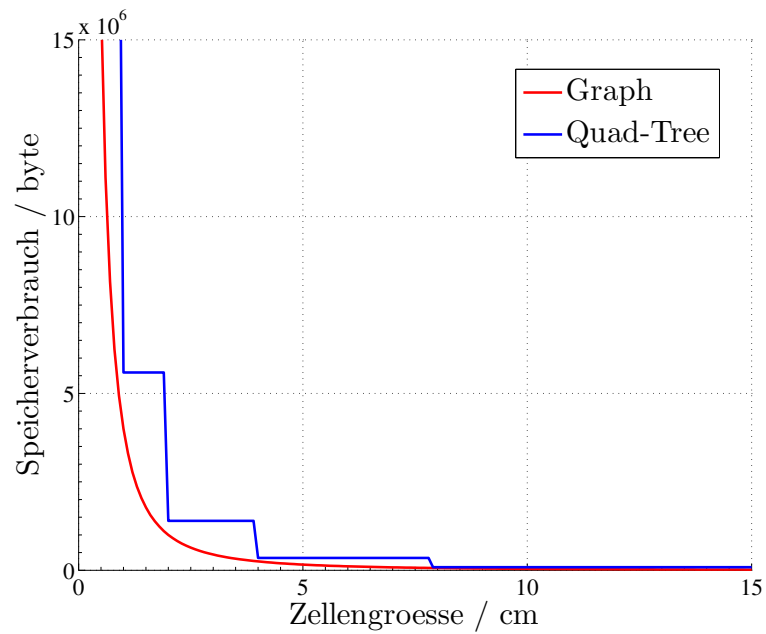


Abbildung 3.4.: Vergleich des Speicherverbrauchs zwischen Graphen und Quad-Tree für eine Fläche von 100m^2

und dem zu vernachlässigendem Unterschied des Speicherverbrauchs bei größeren Zellen, wird der Graph-Tree implementiert.

4. Implementierung

4.1. Überblick

In diesem Kapitel wird zuerst auf die verwendete Hardware (Kapitel 4.2) eingegangen und im Kapitel Softwaredesign 4.3 wird die Implementierung des Mapping Moduls sowie die Integration in die bestehende Software beschrieben. Das Mapping wurde einerseits auf dem bereits vorhandenen Mikrocontrollerboard implementiert und, aufgrund von Speicher- und Geschwindigkeitsproblemen, ebenfalls auf dem PC in die Steuersoftware integriert. Das Microcontrollerboard (Controllerboard Abb. 4.2) ist am Quadrocopter angebracht und wird von diesem getragen. Die Steuersoftware läuft auf dem Steuercomputer (Kontrollcomputer Abb. 4.2), der Bodenstation des Quadrocopters. Zur Evaluierung wurde das Mapping am Steuercomputer durchgeführt.

4.2. Hardwaredesign

Die bestehende Hardware, wie sie in [Müller, 2011] beschrieben ist, wurde vollständig übernommen. Dabei verfügt der Quadrocopter über ein UC3A Controllerboard sowie 12 Ultraschallsensoren. Eine SD-Karte (wie in Abbildung 4.2 dargestellt) mit 128MB wurde zusätzlich implementiert, um genug Speicher für das Mapping bereit zu stellen. Für die Implementierung des Mappings am Steuercomputer (siehe Kapitel 4.3.3) wurden keine Änderungen der Hardware nötig und die SD-Karte wurde in diesem Fall nicht genutzt. Wird das Mapping am Steuercomputer durchgeführt, wird anstelle der Positionsbestimmung über die Sensoren auf ein optisches Tracking System zurück gegriffen. Dies ermöglicht eine genauere Bestimmung der Position im Raum.

Microcontroller Zur Integration der Messdaten in die Map verfügt das Controllerboard über einen AVR32 Microcontroller AT32UC3A0512 mit 66MHz. Dieser besitzt 512kB Flash Speicher

sowie 64kB SRAM [at32uc3, 2012]. Der SRAM des Microcontrollers ist allerdings zu klein, um eine vollständige Karte dauerhaft im Speicher zu halten. Aus diesem Grund wurde eine zusätzliche SD-Karte verwendet. Auch ist die Leistung des Microcontrollers nicht ausreichend das Mapping zwischen den notwendigen Regelungsintervallen durchzuführen.

Sensorik Zur Erfassung der Umgebung werden 12 Ultraschallsensoren vom Typ Devantech SRF02 verwendet, dessen maximale Reichweite 6m beträgt. Die Ultraschallsensoren sind über I²C mit dem Controllerboard verbunden und wie in [Müller, 2011] beschrieben angeordnet. Dabei sind an jedem Ausleger des Quadrocopters drei Sensoren angebracht die jeweils um $27,5^\circ$ versetzt sind. Die Bereiche der Sensoren überschneiden sich durch den 55° großen Öffnungswinkel. Der große Öffnungswinkel stellt ein Problem dar, da aus den Sensordaten nicht bekannt ist, in welchem Winkel zum Sensor das Hindernis erkannt wurde. Dadurch können Zellen falsch als blockiert oder frei in die Map integriert werden (vgl. Abbildung 4.1).

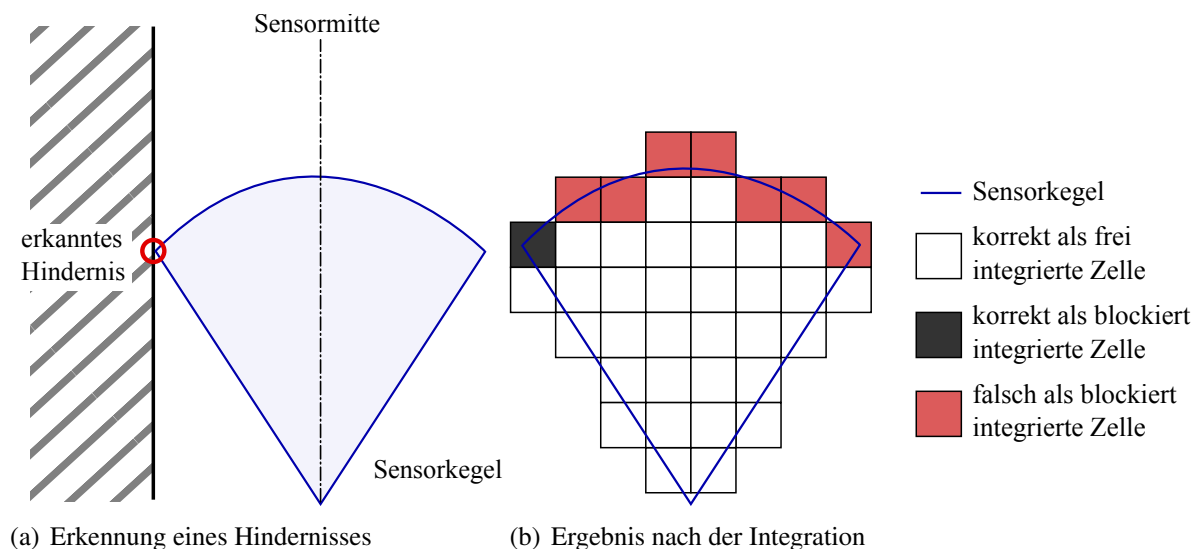


Abbildung 4.1.: Falscherkennung von Zellen

4.3. Softwaredesign

4.3.1. Mapping Software am Quadrocopter

Die Software wurde in die bestehende Kontrollsoftware des Quadrocopters integriert. Dabei wurde darauf geachtet, das Mapping als zusätzliches Modul mit möglichst wenigen Abhängigkeiten zur

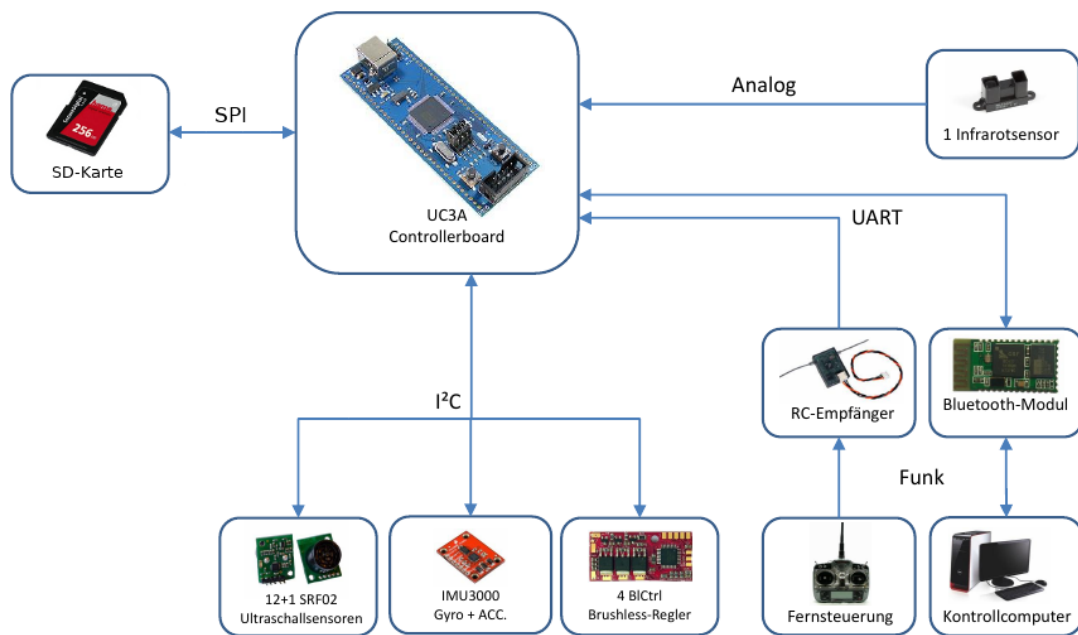


Abbildung 4.2.: Quadcopter Hardwarekonfiguration [Müller, 2011]

bestehenden Software zu implementieren. Dadurch kann das Mapping mit geringem Aufwand bei Bedarf ein- bzw. ausgeschaltet werden. Einzige Abhängigkeit zur Kontrollsoftware besteht in [Müller, 2011] beschriebener Hinderniserkennung. Um eine Konkurrenzsituation zwischen der Hinderniserkennung und dem Mapping beim ansteuern der gemeinsam genutzten Ultraschallsensoren zu vermeiden, werden die Messwerte der Sensoren für das Mapping aus dem Modul der Hinderniserkennung ausgelesen, ohne die Sensoren direkt anzusprechen. Es werden allerdings nicht die gefilterten 24 Sektorwerte, sondern die Rohdaten der Sensoren verwendet. Dies entspricht den 12 Sektoren N, NEE, NEEEE, E, SEEEE, SEE, S, SWW, SWWWW, W, NWWWW und NWW (siehe [Müller, 2011]).

Das Mapping wurde, wie in Abbildung 4.3 dargestellt, implementiert. Dabei werden zuerst die Messdaten vom Hinderniserkennungsmodul ausgelesen, welche in die Positionsbestimmung eingehen. Mit der bekannten Position und dem Messwert werden die vom Sensor erfassten Zellen berechnet. Da der Bresenham-Algorithmus, wie in Kapitel 2.2.1 beschrieben, lediglich Strahlen im 1. Quadranten folgen kann, musste der Algorithmus für die weiteren Quadranten angepasst werden. Die verwendeten Anfangsbedingungen D_{init} und Entscheidungsvariablen Δ können Tabelle A.1

entnommen werden. Für das Controllerboard wurde der Bresenham-Algorithmus gewählt, da er weniger Rechenleistung benötigt (siehe Kapitel 2.2.1).

Nachdem alle Zellen vom Bresenham-Algorithmus bestimmt wurden, werden die Zellen einzeln in die Map integriert und die Baumstruktur rekursiv bis zum Wurzelement aktualisiert. Bei der Integration durch den bayesschen Filter wurde festgestellt, dass sich bei häufiger Beobachtung einer Zelle entweder im blockierte oder freien Zustand, die Wahrscheinlichkeit schnell auf einen Wert nahe 1 bzw. 0 einstellte. Bei der Berechnung des logarithmischen Quotientenverhältnisses kam es somit zu Werten die den Wertebereich der verwendeten Variable überschritten, wodurch diese zu *Infinity* wurde. Auch kann bei der Berechnung von blockierten Zellen durch 0 geteilt werden, was zu *NaN* führte. Um die genannten Probleme zu vermeiden, wurde die Wahrscheinlichkeit vor der Berechnung des logarithmischen Quotientenverhältnisses auf das Intervall $[0.005, 0.995]$ beschränkt.

Ein weiteres Problem stellt der SRAM des Mikrocontrollers dar. Da eine Zelle der Map inklusive aller Pointer auf Kind- und Elternelemente 48B belegt, erwiesen sich die internen 64kB SRAM des Mikrocontrollers bereits für Karten mit geringer Größe als zu klein. Für eine 100m² große Map mit 10cm Zellengröße werden bereits 21844 Zellen benötigt (siehe Tabelle 3.1), was einem Speicherverbrauch von ca. 1024kB entspricht. Dies übersteigt deutlich den internen Arbeitsspeicher. Aus diesem Grund wurde ein zusätzlicher Speicher notwendig. Als Lösung wurde eine SD-Karte in das System integriert (siehe Kapitel 4.3.2). Da sich die Zugriffsgeschwindigkeit der SD-Karte ohne Nutzung des DMA als zu langsam herausgestellt hat, wurde ein Puffer mit variabler Größe implementiert, der die am häufigsten genutzten Zellen im SRAM des Mikrocontrollers speichert. Trotzdem reicht die Geschwindigkeit nicht aus das Mapping zwischen den notwendigen Regelungsintervallen durchzuführen. Aus diesem Grund wurde das Mapping ebenfalls für den PC (siehe Kapitel 4.3.3) implementiert.

4.3.2. SD Karte

Die Implementierung der SD-Karte erwies sich aus den vorgenannten Gründen (siehe Kapitel 4.3.1) als notwendig. Um die SD-Karte so flexibel wie möglich zu halten, wurde es als eigenes Modul implementiert. Angebunden ist die SD-Karte wie in Abbildung 4.2 dargestellt über den SPI-Bus. Das zur Verfügung stehende Software Framework des AT32 unterstützt dabei SD-Karten die eine Größe von maximal 1GB haben. Als Dateisystem auf der SD-Karte wird *FAT12*, *FAT16* und

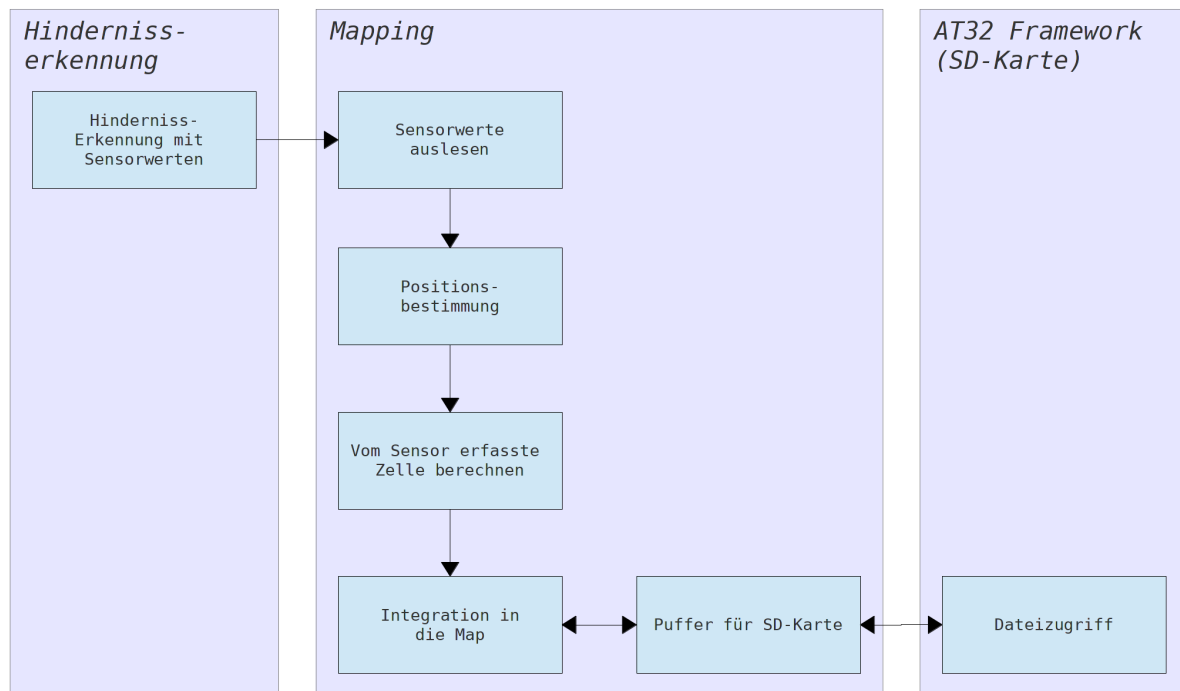


Abbildung 4.3.: Mapping Software Modul am Quadrocopter

FAT32 unterstützt [atmel, 2007]. Durch die Nutzung mehrerer Navigatoren für die File Allocation Table ist die SD-Karte nicht nur für das Mapping nutzbar, sondern ebenfalls für andere Module, die in Zukunft ebenfalls die SD-Karte nutzen möchten.

Der Zugriff auf die SD-Karte selbst erfolgt über die durch das AT32 Framework in *file.h* und *navigation.h* bereitgestellten Funktionen. Das SD-Kartenmodul übernimmt ausschließlich die Initialisierung der SD-Karte. Es können neue Module, die die SD-Karte nutzen wollen, einfach hinzugefügt werden. Für jedes Modul wird ein Eintrag in einer Enumeration sowie ein Arbeitsverzeichnis angegeben. Bei der Initialisierung wird ein Navigator für jedes Modul erstellt und das entsprechende Arbeitsverzeichnis angelegt, falls es nicht existiert. Der Zeiger wird nach dem Anlegen des Verzeichnisses dorthin gesetzt. Über die Funktion `nav_select()` kann, nach der Initialisierung, mit dem Namen des Moduls als Parameter, der dem Modul zugeordnete Navigator ausgewählt werden. Alle anderen Zeiger bleiben so an ihrer bisherigen Stelle bestehen und es kann leicht zwischen den Modulen gewechselt werden. Da jedes Modul so über einen eigenen Zeiger verfügt, werden Konflikte beim ändern des Navigators vermieden. Es ist jedoch trotzdem möglich seinen eigenen Zeiger in fremde Arbeitsverzeichnisse zu setzen oder den Zeiger eines fremden Moduls über `nav_select()` auszuwählen.

4.3.3. Mapping Software am PC

Da die Ausführungsgeschwindigkeit des Mappings direkt auf dem Mikrocontroller nicht ausreicht, wurde es in die Steuersoftware integriert. Die Messdaten der Sensoren werden über die bestehende Funkstrecke an den Steuercomputer gesendet und das Mapping lokal am PC durchgeführt. Dabei werden dieselben Sensorsektoren genutzt wie in Kapitel 4.3.1 bereits erwähnt.

Da ohne Position keine Map erstellt werden kann, stellt die Positionsbestimmung im Indoor-Bereich ein großes Problem dar. Die Positionsbestimmung über die Ultraschallsensoren ist ungenauer und verursacht so mehr Fehler in der Map. Aus diesem Grund wurde als erste Änderung die Positionsbestimmung angepasst. Am Steuercomputer stehen die Positionsdaten eines optischen Tracking-Systems zur Verfügung. Anstelle der aus den Sensordaten berechneten Position wird am PC die Position des optischen Trackings für das Mapping verwendet. Dadurch gehen die Messungenauigkeiten der Sensoren nicht doppelt, als Position und Hinderniserkennung, in den Mappingalgorithmus ein. Zudem ist das optische Tracking System genauer als die verwendeten Ultraschallsensoren. Da das optischen Tracking System die Position liefert und die Position nicht anhand der Messungen aus der Map bestimmt wird, grenzt es sich hiermit von einem SLAM (Simultaneous Localization and Mapping) Verfahren ab.

Eine weitere Änderung ist die Bestimmung der vom Sensor erfassten Zellen. Es wurde anstelle des beschriebenen Bresenham Algorithmus das in Kapitel 2.2.1 beschriebene Sektormodell genutzt. Damit nicht für alle Zellen der Map geprüft werden muss ob sich die Zelle im Sensorkegel befindet, werden zuerst die Referenzpunkte u , v und w berechnet. Die Position des Sensors x ist durch die Position des Quadrocopters und dem Versatz gegeben. Aus den Minima sowie Maxima der Koordinaten der Punkte, kann eine Begrenzung um den Sektor aufgespannt werden (vgl. Abb. 4.4 und Algorithmus 4.1 Zeile 2-5). Die Minima dienen hierbei als Anfangswerte der Schleifen, die die Begrenzung durchlaufen (Zeile 7 und 8). Die Maxima dienen als Abbruchbedingung der Schleifen. Alle Zellen die außerhalb der Begrenzung liegen, können nicht im Sensorsektor liegen und müssen nicht beachtet werden. Es werden lediglich die Zellen aktualisiert, die innerhalb der Begrenzung liegen (siehe Algorithmus 4.1 Zeile 9).

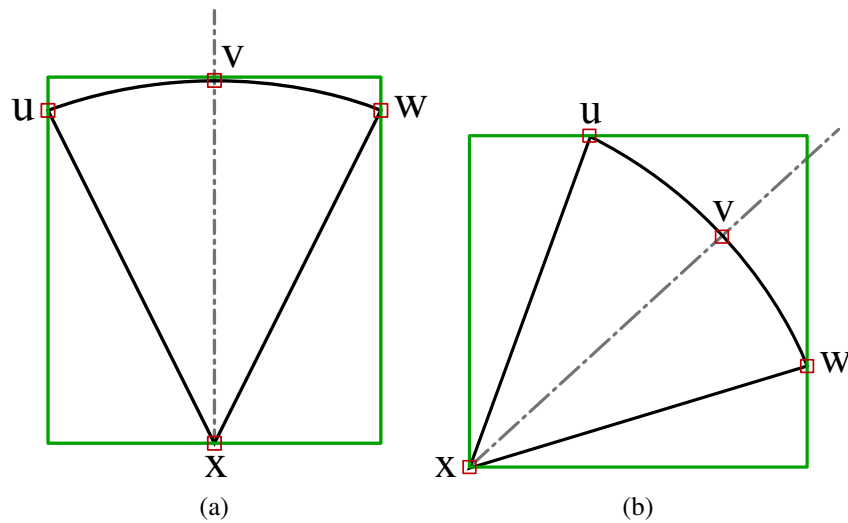


Abbildung 4.4.: Implementiertes Sektormodell der Ultraschallsensoren

```

1 sector_model( $x_t$ ,  $u_t$ ,  $v_t$ ,  $w_t$ ,  $z_t$ )
2   int bottomX = min( $x_{t,x}$ , min( $u_{t,x}$ , min( $v_{t,x}$ ,  $w_{t,x}$ )));
3   int bottomY = min( $x_{t,y}$ , min( $u_{t,y}$ , min( $v_{t,y}$ ,  $w_{t,y}$ )));
4   int topX    = max( $x_{t,x}$ , max( $u_{t,x}$ , max( $v_{t,x}$ ,  $w_{t,x}$ )));
5   int topY    = max( $x_{t,y}$ , max( $u_{t,y}$ , max( $v_{t,y}$ ,  $w_{t,y}$ )));
6
7   for (int i = bottomX; i <= topX; i++) {
8     for (int j = bottomY; j <= topY; j++) {
9       integrate( $m_{i,j}$ , inverse_sensor_model( $m_{i,j}$ ,  $x_t$ ,  $z_t$ ));
10    }
11  }

```

Algorithmus 4.1: Implementiertes Sektorenmodell

Da am Steuercomputer zudem ausreichend Speicher zur Verfügung steht, muss nicht auf externe Speicher wie SD-Karten zurückgegriffen werden. Die Steuersoftware übernimmt zudem die Anzeige der aktuellen Map, unabhängig davon ob das Mapping online, auf dem Mikrocontroller, oder offline am Steuercomputer ausgeführt wird. Das Mapping am Steuercomputer besteht aus den folgenden vier Hauptbestandteilen wie Abbildung 4.5 zeigt.

Quad-Tree Der Quad-Tree bildet die Baumstruktur im Speicher ab. Im Offline Betrieb kümmert sich der Quad-Tree zusätzlich um die Integration der Messwerte nach den in Kapitel 3 vorgestellten Verfahren. Im Online Betrieb dient der Quad-Tree lediglich der Darstellung im GUI-Controller.

GUI-Controller Der GUI-Controller übernimmt die Anzeige der aktuellen Map in der Steuerungssoftware. Er kümmert sich ebenfalls um die Übernahme der einstellbaren Mapping-Parameter. Dazu gehören:

- Größe der gesamten Map
- Größe der einzelnen Zellen
- A priori Wahrscheinlichkeit des Bayes-Filters $p(\mathbf{m}_i)$
- Bayes-Wahrscheinlichkeit $p(\mathbf{m}_i|z_t, x_t)$ für frei und blockiert beobachtete Zellen
- Inkrementierungsvariablen für frei und blockiert beobachtete Zellen

Datenreceiver Im Datenreceiver werden alle für das Mapping notwendigen Daten entgegen genommen und in eine gemeinsame Form gebracht. Wird das Mapping im Online Betrieb auf dem Controllerboard ausgeführt, werden die Zellen vom Controllerboard an den Kontrollcomputer gesendet, von *Big-Endian* nach *Little-Endian* konvertiert und in einer Baumstruktur gespeichert. Die Anwendung des Bayes- und Zählverfahrens erfolgt am Controllerboard. Die übertragenen Zellen enthalten die beiden Zählvariablen c_{occ} und c_{free} , die Bayes-Wahrscheinlichkeit der Zelle und die Zellposition und Zellabmessungen.

Im Offline Betrieb nimmt der Datenreceiver die rohen Messwerte der Ultraschallsensoren entgegen und kümmert sich um die Synchronisation mit dem optischen Tracking-System, damit die Sensordaten nicht mit der falschen Position in die Map einfließen. Die Integration der Messung in die Map erfolgt dabei am Steuercomputer durch beide Verfahren. Die Positions kann bei Bedarf, wenn keine andere Positionsbestimmung zur Verfügung steht, auch per Hand eingegeben werden.

Ebenfalls können auch Map-Dateien von der SD-Karte geladen werden, da die Steuerungssoftware für Map-Dateien dasselbe Format wie das Controllerboard verwendet. Beim Einlesen am Steuercomputer ist jedoch eine Änderung der Endianess der Daten notwendig. Es können auch Simulationsdateien mit generierten Positions- und Sensordaten eingelesen werden aus denen durch das implementierte Mapping-Verfahren eine Karte erstellt wird.

Datenexport Der Datenexport ermöglicht das Speichern von Map-Dateien der aktuellen Map in der korrekten Endianess, die vom Controllerboard wieder eingelesen werden können. Zudem kann die aktuell angezeigte Map ebenfalls als Bitmap exportiert werden.

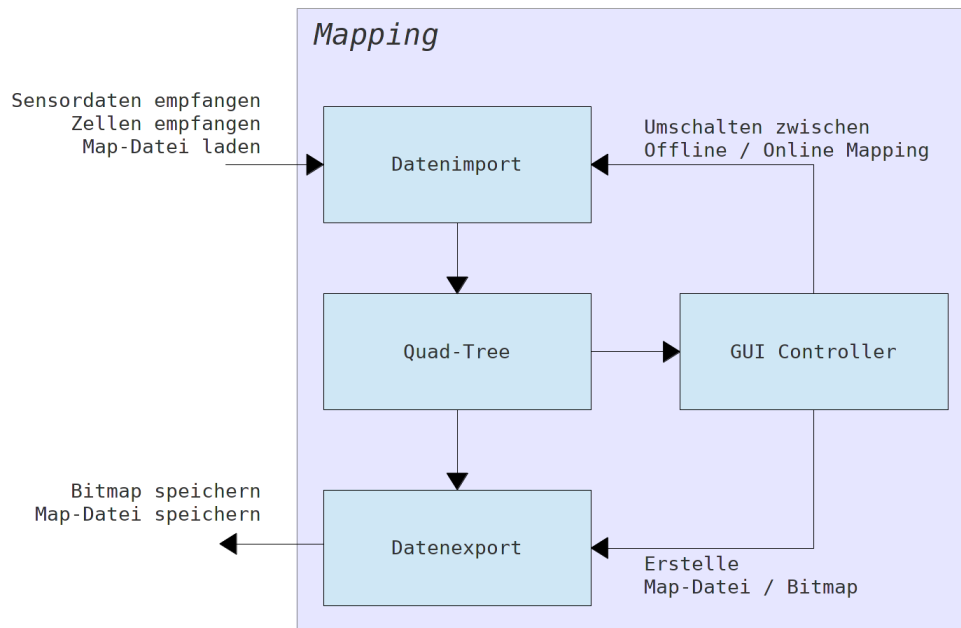


Abbildung 4.5.: Mapping Software Modul

4.3.4. GUI am PC

Die grafische Benutzeroberfläche für das Mapping wurde in die bestehende Qt-GUI des AQopterI8-Projekts integriert. Dabei stehen zwei Reiter (Tabs) zur Auswahl. Im Tab *Darstellung* (vgl. Abbildung 4.6(a)) kann die Ansicht der aktuellen Map verändert werden. Er zeigt auch aktuelle Informationen bezüglich der Map an. Der Tab *Einstellungen* (vgl. Abbildung 4.6(b)) dient der Einstellung von Parametern die die Mapping-Verfahren, Bayes- und Zählverfahren, beeinflussen. Die Map- und Zellengröße kann ebenfalls verändert werden.

Tab Darstellung

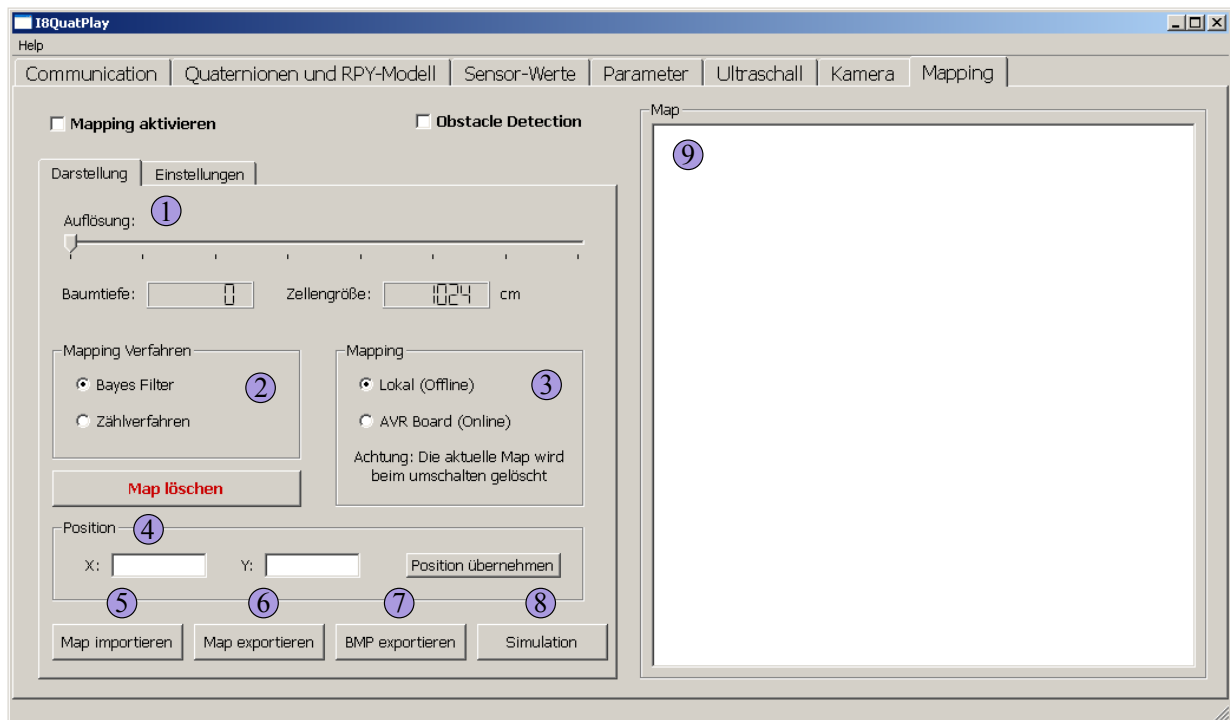
1. Dieser Schieberegler dient der Einstellung der angezeigten Auflösung. Da ein Quad-Tree implementiert wurde, kann die Auflösung durch einstellen der Baumtiefe verändert werden. Unter der Einstellung befindet sich die Anzeige der mit der Auflösung verbundenen Baumtiefe und Zellengröße.
2. Beide Integrationsverfahren werden parallel ausgeführt. Der Menüpunkt ermöglicht das Umschalten der Map-Anzeige. Dabei kann zwischen der Map des Bayes-Filters und des Zählverfahrens gewechselt werden.

3. Hiermit kann das Mapping zwischen lokalem Mapping am PC und dem Mapping auf dem Controllerboard umgeschaltet werden. Wird Offline gewählt, wird das Bayes- und Zählverfahren am Kontrollcomputer durchgeführt. Im Online-Betrieb führt das Controllerboard am Quadrocopter die Verarbeitung durch.
4. X und Y sind die Anzeige der aktuellen Position. Falls keine andere Positionsbestimmung möglich ist, kann durch den Button *Positions übernehmen*, ein in die Felder X und Y eingegebene Position auch per Hand in das Mapping-Verfahren übernommen werden.
5. Dieser Button lädt eine Map-Datei von einer SD-Karte oder Festplatte.
6. Dieser Button erstellt eine Map-Datei und speichert diese ab.
7. Hiermit wird die aktuell angezeigte Map als Bitmap exportiert.
8. Dieser Button lädt eine Simulationsdatei und verwendet die darin gespeicherten Positions- und Sensordaten für das Mapping.
9. In diesem Fenster wird die Map angezeigt. Diese Anzeige kann durch die Auflösung (1) und das gewählte Verfahren (2) verändert werden.

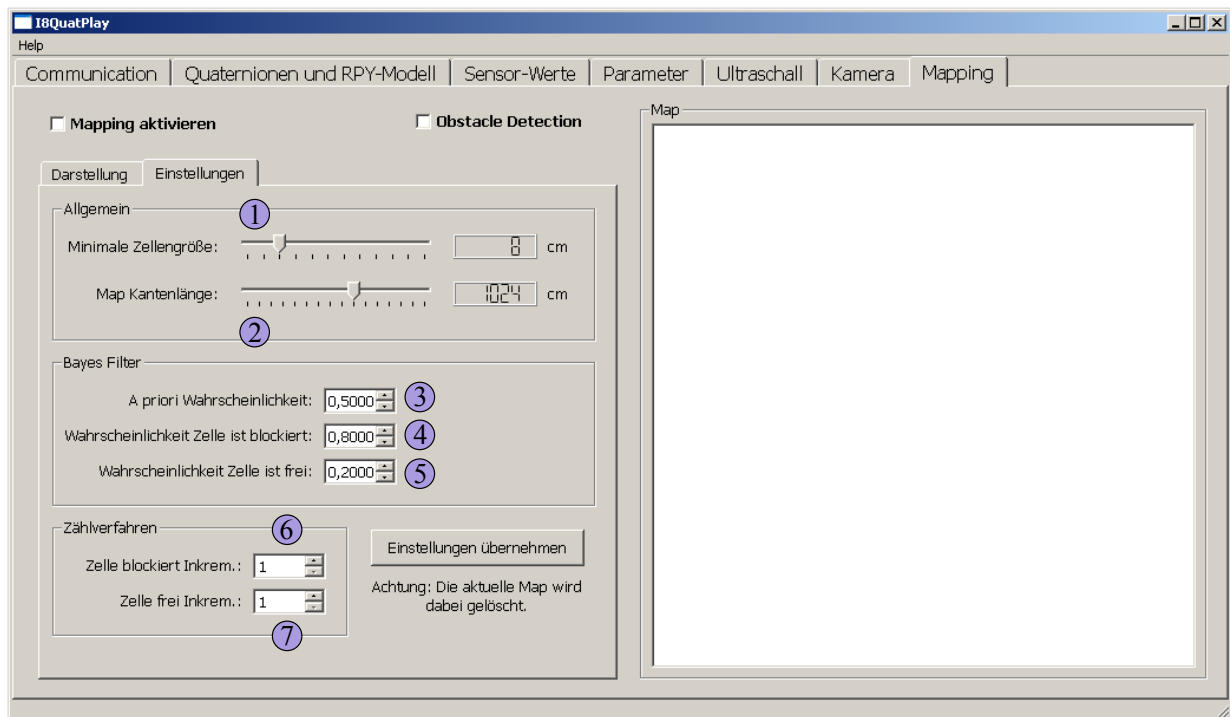
Tab Einstellungen

1. Hier kann die Gesamtgröße der Map eingestellt werden. Je größer die Map wird, desto tiefer wird die Baumstruktur.
2. Durch diesen Schieberegler kann die vom Mapping verwendete Zellengröße eingestellt werden. Der eingestellte Wert bezieht sich auf die minimale Größe einer Zelle. Eine Zelle mit der eingestellten Größe wird nicht weiter unterteilt. Je kleiner die Zelle wird, desto tiefer wird der Quad-Tree, da häufiger unterteilt werden muss.
3. Dieser Wert stellt die a priori Wahrscheinlichkeit des bayesschen Filters ein. Neu angelegte Zellen werden mit der a priori Wahrscheinlichkeit initialisiert.
4. Mit diesem Wert kann die Wahrscheinlichkeit für blockierte Zellen eingestellt werden. Dieser Wert muss im Intervall $[0.5, 1.0]$ liegen. Liegt dieser Wert unter 0.5, werden als blockiert erkannte Zellen als frei in die Map integriert.

5. Hiermit wird die Wahrscheinlichkeit für freie Zellen geändert. Diese muss im Intervall $[0.0, 0.5]$ liegen. Bei einem Wert über 0.5 werden auch als frei erkannte Zellen als blockiert integriert.
6. Dieser Wert ändert die Inkrementierungsvariable für blockierte Zellen. Die Inkrementierungsvariable wird zur Erhöhung der Zählvariable c_{occ} genutzt.
7. Mit diesem Wert wird die Inkrementierungsvariable für freie Zellen eingestellt. Diese wird zur Erhöhung der Zählvariable c_{free} verwendet.



(a) Qt-Gui Mapping-Tab mit Optionen zur Darstellung der Map



(b) Qt-Gui Mapping-Tab mit Optionen der Mapping Verfahren

Abbildung 4.6.: Qt-GUI

5. Evaluierung

5.1. Überblick

Bei der Evaluierung wurde versucht, optimale Parameter für die beschriebenen Integrationsverfahren zu finden, um eine möglichst aussagekräftige und detaillierte Map zu erstellen. Dabei wurde die Reichweite des Sensors, der Öffnungswinkel des Sensor, die Bayes Wahrscheinlichkeit, die Inkrementierungsvariablen für das Zählverfahren und die Zellengröße unabhängig voneinander angepasst und die Auswirkungen auf das Mapping untersucht. Dabei wurden pro Position 10 Messwerte der 12 Sensoren in die bestehende Map eingearbeitet und der Quadrocopter danach weiterbewegt. Zur Integration wurden beide Verfahren parallel genutzt. Der Quadrocopter wurde nicht, wie im realen Fall, kontinuierlich durch die Umgebung bewegt, sondern nur an festgelegten Position aufgestellt. Es kann vorkommen, dass der Quadrocopter sich auch für eine gewisse Zeit an einer Position aufhält. Um die kontinuierliche Bewegung und auch das warten an einer Position möglichst nachzubilden, wurden 10 Messwerte pro Position verwendet. Außerdem wurde ein Vergleich zwischen Bayes- und Zählverfahren durchgeführt. Zur Evaluierung wurden ausschließlich die Rohdaten der Sensoren verwendet. Eine Sensorfusion wurde nicht verwendet.

5.2. Optimierung

Die Optimierung der Parameter wurde in einer einfachen Umgebung (vgl. Abbildung 5.1), einem rechteckigen Raum, durchgeführt. Der Raum hat die Länge $l = 401\text{cm}$ und eine von Breite $b = 232\text{cm}$ bzw. $b = 202\text{cm}$. Dies hilft Sensorfehler gering zu halten. Somit sind auch die Ergebnisse besser vergleichbar. Die Optimierung wurde durchgeführt ohne den Sensorversatz gegenüber der Position des Quadrocopters zu berücksichtigen.

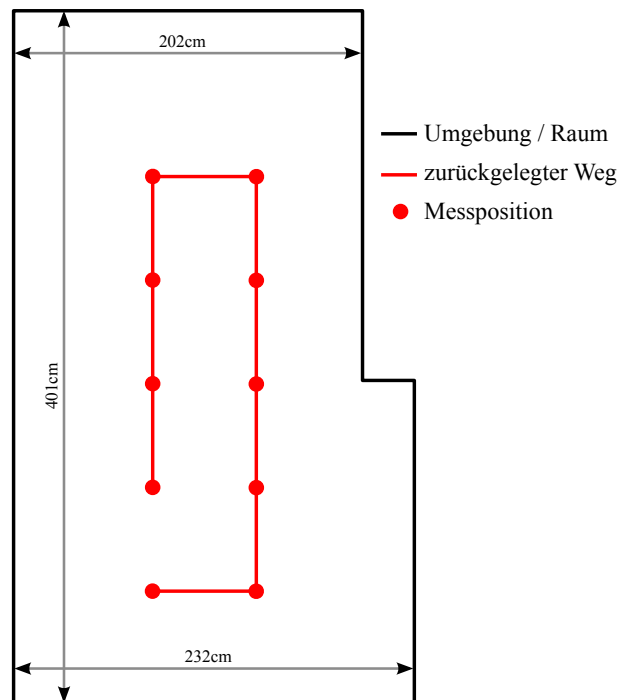


Abbildung 5.1.: Zu kartografierende Umgebung

5.2.1. Reichweite des Sensors

Hier wurde untersucht wie sich die Begrenzung der Sensorreichweite auf das Mapping auswirkt, da mit steigendem Abstand der gemessene Sektor und der als blockiert betrachtete Kreisbogenschnitt größer werden. Dabei wurde die maximale Sensorreichweite immer weiter reduziert. Alle weiteren Parameter (siehe Tabelle A.2) wurden konstant gehalten.

Aus den Karten in Tabelle A.7 ist ersichtlich, dass sich die Beschränkung der Sensorreichweite positiv auf das Mapping auswirkt. Mit sinkender Reichweite nehmen auch die Artefakte außerhalb der Map zunehmend ab. Dies liegt am kleiner werdenden Kreissektor des Sensors, der in die Karte integriert wird. Dadurch wird der Bereich der, möglicherweise als fehlerhaft blockiert betrachteten, Zellen verkleinert (vgl. Abbildung 4.1). Der große Sektor hat zudem die negative Eigenschaft, auf größere Entfernungen, bereits als blockiert erfasste Zellen wieder in einen undefinierten oder gar freien Zustand zurück zu bringen, was besonders auf der linken Hälfte der Map 5.2(a) und 5.2(b) deutlich wird. Diese Problem wird durch einen zu flachen Winkel zwischen Hindernis und Sensor verursacht, da der Sensor, bedingt durch die Winkelabhängigkeit (siehe [Müller, 2011]), Fehlmessungen verursacht.

Eine weitere künstliche Verkürzung der Sensorreichweite auf unter 150cm erschien nicht sinnvoll, da sonst zu dicht an Hindernisse heran geflogen werden müsste, und wurde deswegen nicht weiter verfolgt. Aus diesem Grund wird, für die verwendete Konfiguration und Umgebung, eine Sensorreichweite von 150cm als optimaler Wert aufgefasst.

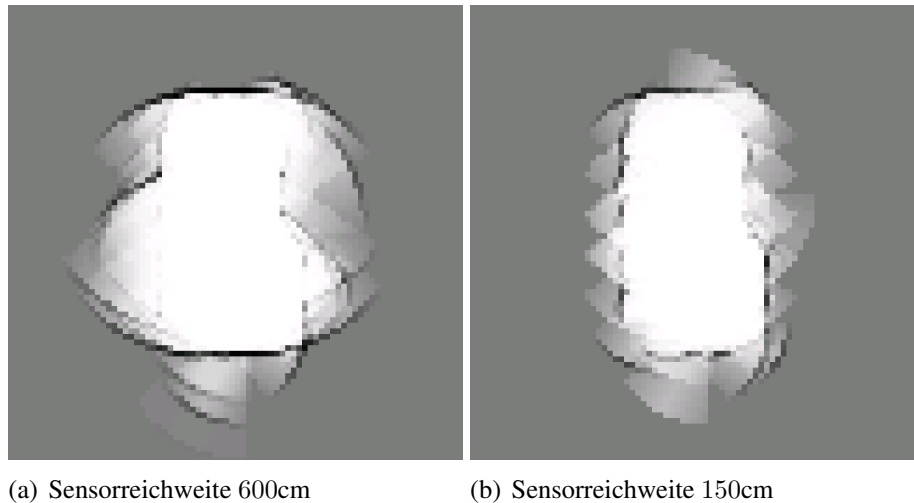


Abbildung 5.2.: Anpassung der Sensorreichweite

5.2.2. Sensoröffnungswinkel

In diesem Kapitel wird die Auswirkung des Sensoröffnungswinkels auf das Mapping untersucht. Dabei ist zu beachten, dass der Sensor selbst nicht beeinflusst wurde. Die Änderung des Öffnungswinkels wurde lediglich softwareseitig durchgeführt. Da der Öffnungswinkel die Größe des zu integrierenden Sektors, und damit die Qualität der Map, stark beeinflusst, wurden die Auswirkungen des Winkels auf das Mapping Verfahren evaluiert. Die verwendeten Parameter sind in Tabelle A.3 aufgeführt.

Dabei zeigt Tabelle A.8 das sich, auch bei starker Reduzierung des Öffnungswinkels, kaum Auswirkungen auf das Verhalten der Mapping Verfahren ergeben. Bei zu kleinen Öffnungswinkeln bilden sich zudem deutliche Bereiche, in denen der Sensor die Zellen nicht oft genug bzw. gar nicht erfassen konnte. Diese lassen somit keine klare Aussage mehr zu, ob die Zelle frei oder blockiert ist. Im Zählverfahren nahmen bis 35° die Artefakte außerhalb der Map jedoch leicht ab. Diese Artefakte bilden sich durch fehlerhafte Messungen, in denen der Sensor zu weit gegenüber

dem Hindernis gekippt war. Auch bilden sich Artefakte innerhalb der Map durch das in Kapitel 4.2 beschriebene Problem.

Der geringe Einfluss des Öffnungswinkels lässt sich durch die bereits verkürzte Sensorreichweite erklären. Da mit kürzeren Sensorreichweiten auch der Kreisbogenausschnitt kleiner wird, nimmt der Einfluss des Sensoröffnungswinkels ebenfalls ab. Eine Verringerung des Öffnungswinkels unter 35° erweist sich dadurch im Zusammenhang mit einer Reduzierung der Sensorreichweite als wenig praxistauglich. Ein Winkel von 35° stellt einen optimalen Kompromiss zwischen der vom Ultraschallsensor erfassten Zellenmenge, über die eine Aussage getroffen werden kann, und der Artefaktbildung in der Nähe von erkannten Hindernissen dar.

5.2.3. Zellengröße

Weiterhin wurde die Änderung der Zellengröße und die damit verbundenen Auswirkungen auf das Mapping untersucht. Dabei wurde bei einer Zellengröße von 16cm begonnen und mit jedem Schritt, bis hin zu 2cm, die Größe halbiert. Die weiteren Parameter sind in Tabelle A.4 aufgeführt.

Zwar sind, wie Abbildung 5.3(a) zeigt, bei einer Größe von 16cm die Wände stark ausgeprägt, die Map ist im Gegenzug dafür aber sehr grob. Mit sinkender Zellengröße steigt der Detailgrad der Map an. Die Wände sind mit sinkender Zellengröße jedoch immer schlechter definiert. Die Wahrscheinlichkeit der Zellen, die die Wände repräsentieren, sinkt (vgl. Tabelle A.9). Bei einer Größe von 2cm (Abbildung 5.3(b)) ist die Wand im Bayes-Verfahren bereits kaum mehr von der undefinierten Umgebung zu unterscheiden. Dies lässt sich darauf zurück führen, dass Sensorschwankungen häufiger in dieselbe Zelle fallen je größer die Map aufgelöst ist. Bei kleiner Zellengröße können bereits niedrige Schwankungen im Messwert um wenige Zentimeter ausreichen, um Nachbarzellen zu erreichen. Somit wurden mehr unterschiedliche Zellen als blockiert beobachtet, die zudem weniger stark ausgeprägt wurden. Allerdings wurde die Abweichung der Abmessung der Map gegenüber der tatsächlichen Umgebung mit sinkender Zellengröße ebenfalls kleiner, was sich mit der höheren Detailgenauigkeit erklären lässt.

Zudem wurde mit jeder Halbierung der Zellengröße die Anzahl der Zellen quadriert, was sich negativ auf die Sample-Rate des Mappings auswirkt. So wurde zur Evaluierung einer Zellengröße von 2cm der Zeitraum zwischen den Messungen auf 1500ms erhöht, im Gegensatz zu 270ms, die für die restliche Evaluierung verwendet wurde. Der Wert von 1500ms ist absichtlich hoch geschätzt worden, um dem Steuercomputer ausreichend Zeit für die Integration zu geben, da es

sonst zu Verbindungsabbrüchen in der Bluetooth-Funkverbindung kam. Dieser Wert kann durch Optimierung der Software noch reduziert werden. Insgesamt erwies sich eine Zellengröße von 4cm als ausreichend schnell und detailliert.

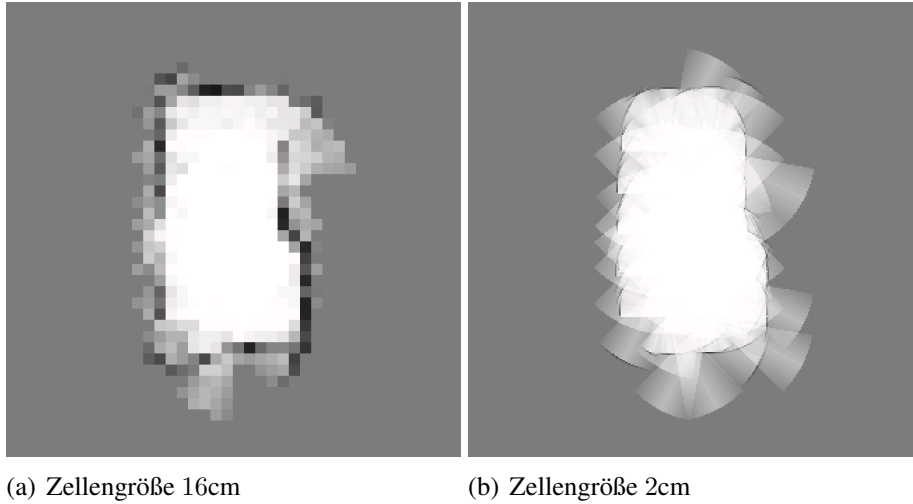


Abbildung 5.3.: Anpassung der Zellengröße

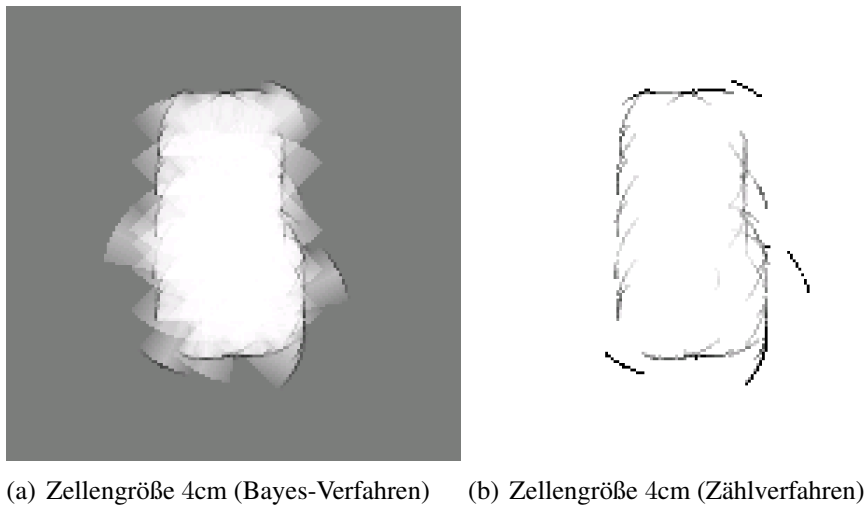


Abbildung 5.4.: Optimaler Wert der Zellengröße

5.2.4. Bayes Wahrscheinlichkeiten

In diesem Kapitel wurde die Optimierung der Bayes-Wahrscheinlichkeiten $p(\mathbf{m}_i | z_t, x_t)_{free}$ für freie und $p(\mathbf{m}_i | z_t, x_t)_{occ}$ für blockierte Zellen evaluiert. Die anderen Parameter können Tabelle

A.5 entnommen werden. Bei der Optimierung lag das Ziel darin, die Wahrscheinlichkeiten so zu wählen, dass eine gute Map mit den verbauten Ultraschallsensoren erstellt werden kann.

Dabei gilt, je näher die Werte an 0,5 liegen, umso träger wird das Mapping-Verfahren bzw. desto geringer ist der Einfluss der aktuellen Messung auf die bereits erstellte Map (vgl. Abbildung A.3). Zu Beginn wurden für die ersten vier Evaluierungen $p(\mathbf{m}_i|z_t, x_t)_{free}$ und $p(\mathbf{m}_i|z_t, x_t)_{occ}$ synchron geändert, sodass die Summe 1 ergibt. Da mehrere Messungen pro Position gemacht wurden, erwiesen sich Werte mit $p(\mathbf{m}_i|z_t, x_t)_{free} < 0,4$ und $p(\mathbf{m}_i|z_t, x_t)_{occ} > 0,6$ als schlecht (vgl. Tabelle A.10). Durch den großen Sensorkegel wurden dadurch häufig Zellen, die bereits korrekt als blockiert beobachtet wurden, durch nachfolgende Messungen, von anderen Positionen aus, als frei markiert. Dies lässt sich in Abbildung 5.5(a) und 5.5(b) unten links, sowie auf der linken Seite in Abbildung A.4 beobachten. Aus diesem Grund wurden weitere trägere Werte getestet (vgl. Abb. A.3), die bessere Ergebnisse lieferten, wie Abbildung 5.6 zeigt.

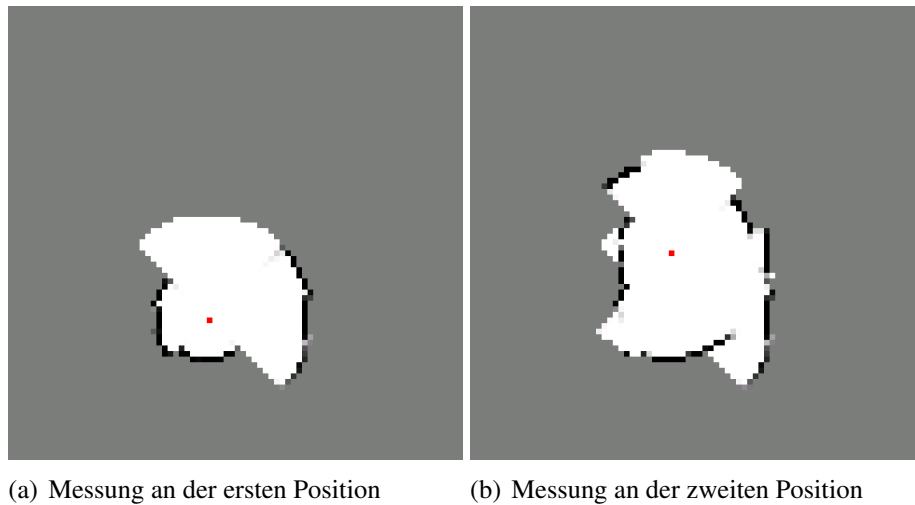


Abbildung 5.5.: Zwei Messschritte¹ mit $p(\mathbf{m}_i|z_t, x_t)_{free} = 0,2$ und $p(\mathbf{m}_i|z_t, x_t)_{occ} = 0,8$

Da blockierte Zellen zwar erkannt wurden, diese jedoch durch einen zu niedrigen Wert für $p(\mathbf{m}_i|z_t, x_t)_{free}$ an einigen Stellen $p(\mathbf{m}_i)$ wieder deutlich unter 0,5 sank, wurde in drei Evaluierungen $p(\mathbf{m}_i|z_t, x_t)_{free}$ leicht träger eingestellt als $p(\mathbf{m}_i|z_t, x_t)_{occ}$. Dies hat zur Folge, dass Zellen schneller als blockiert erkannt werden können und mehr Messungen notwendig sind, diese wieder als frei zu markieren. Mit diesen Werten konnte ein besseres Ergebnis erzielt werden (vgl. Abbildung 5.7). Es zeigen sich jedoch immer noch kleinere Ungenauigkeiten in der Map,

¹Position rot gekennzeichnet

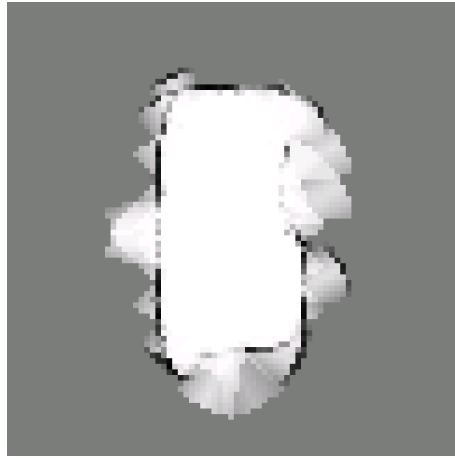


Abbildung 5.6.: Map mit $p(\mathbf{m}_i|z_t, x_t)_{free} = 0,45$ und $p(\mathbf{m}_i|z_t, x_t)_{occ} = 0,55$

die jedoch durch Anpassung der Bayes-Wahrscheinlichkeiten nicht beseitigt werden konnten. Für die gegebene Evaluierung erwiesen sich träge Werte von $p(\mathbf{m}_i|z_t, x_t)_{free} = 0,4775$ und $p(\mathbf{m}_i|z_t, x_t)_{occ} = 0,55$ als optimal.

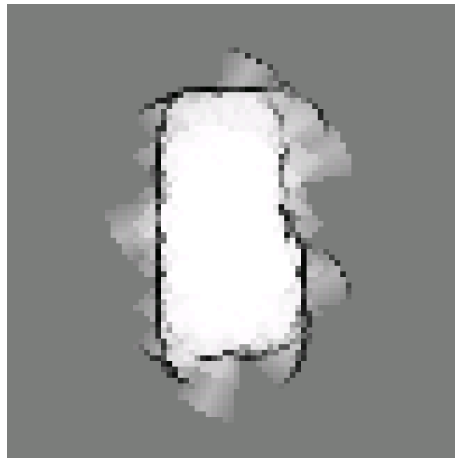


Abbildung 5.7.: Map mit $p(\mathbf{m}_i|z_t, x_t)_{free} = 0,4775$ und $p(\mathbf{m}_i|z_t, x_t)_{occ} = 0,55$

5.2.5. Zählvariablen

Im folgenden Kapitel wurden die Inkrementierungsvariablen für das Zählverfahren optimiert. Dabei wurde das Verhältnis der Inkrementierungsvariablen $\frac{m_{occ}}{m_{free}}$ Schritt für Schritt verändert und das Ergebnis untersucht. Weiterhin wurden die in Tabelle A.6 aufgeführten Werte verwendet.

Für ein Verhältnis mit $\frac{m_{occ}}{m_{free}} > 1$ konnte keine deutliche Verbesserung oder Verschlechterung des Zählverfahrens festgestellt werden. Auch die Vergrößerung der Steigung m durch größere

Inkrementierungsvariablen für die lineare Verteilung (Gleichung 3.4) brachte keine Verbesserung des Verfahrens (vgl. Tabelle A.11). Für $\frac{m_{occ}}{m_{free}} \leq 1$ wurden die Zellen mit einer niedrigeren Wahrscheinlichkeit belegt und die Umrissse der Map erscheinen so weniger gut definiert (vgl. Abbildung 5.8), weswegen keine weitere Verringerung von $\frac{m_{occ}}{m_{free}}$ mehr durchgeführt wurde. Ein Verhältnis von $\frac{m_{occ}}{m_{free}} > 1$ wurde bevorzugt, da sich hierdurch besser definierte Karten ergaben. Die Werte $m_{free} = 1$ und $m_{occ} = 2$ werden aus diesen Gründen als Optimum für die gegebene Konfiguration betrachtet.

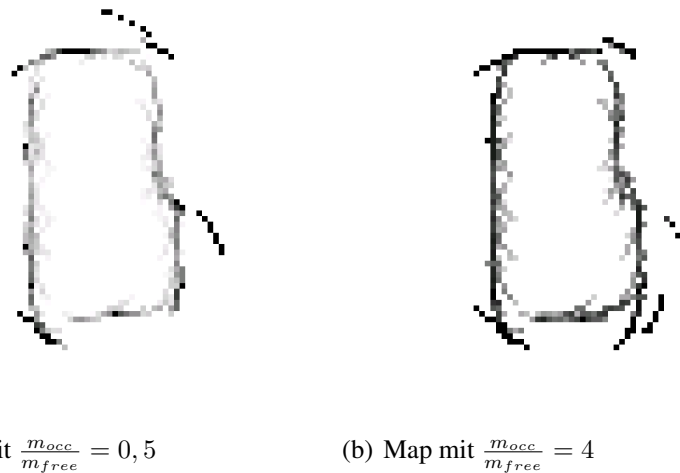


Abbildung 5.8.: Maps erstellt mit dem Zählverfahren

5.3. Vergleich zwischen Bayes- und Zählverfahren

In diesem Kapitel wurden das Bayes- und Zählverfahren miteinander verglichen und die Vor- und Nachteile evaluiert. Hierfür wurden die im vorgenannten Kapitel optimierten Parameter (siehe Tabelle A.12) verwendet. Es wurde die einfache Umgebung, wie in Kapitel A.3, verwendet sowie eine komplexere Umgebung (siehe Abbildung A.5). In der Evaluierung der komplexen Umgebung wurde der Versatz der Sensoren, gegenüber der Quadrocopterposition, berücksichtigt. Das optische Tracking System konnte den Quadrocopter in der komplexen Umgebung, im Bereich oben rechts und unten links, nur schlecht bis gar nicht erkennen. Aus diesem Grund wurde der Quadrocopter dort nicht platziert und es kommt dort zu Ungenauigkeiten. Deswegen ist die Umgebung an diesen

Stellen nicht gut in die Map abgebildet worden, was sich in Abbildung 5.9 sowohl im Bayes- als auch im Zählverfahren zeigt.

Der Aufbau der Map durch das Bayes-Verfahren lässt sich in Abbildung A.7(a) bis A.7(o) für das Bayes-Verfahren, sowie Abbildung A.8(a) bis A.8(p) für das Zählverfahren, nachvollziehen. Dabei sind deutlich die durch den Sensorkegel entstehenden Rundungen in den Wänden zu erkennen. Der Aufbau der Map des Zählverfahrens läuft analog zum Bayes-Verfahren.

Das Bayes-Verfahren zeigt in der simplen Umgebung eine leicht bessere Map als das Zählverfahren, da hier weniger Artefakte innerhalb der Map entstehen (siehe Abbildung A.6). Die Wände sind in beiden Verfahren ähnlich gut definiert. Ansonsten unterscheiden sich die beiden Verfahren in der einfachen Umgebung kaum. Wie in Abbildung 5.9 dargestellt, weist in der komplexen Umgebung die Map für das Zählverfahren eine bessere Wanderkennung auf. Dies lässt sich durch die schnelle Anpassung des Bayes-Verfahrens, aufgrund des logarithmischen Quotientenverhältnisses (vgl. Abbildung A.3), an die zuletzt durchgeführten Messungen erklären. Die freien Flächen werden von beiden Verfahren gut erkannt. Das Zählverfahren hat leicht besser definierte Wände. Dies zeigt sich in der komplexen Umgebung im unteren und an der rechten Seite. In der einfachen rechteckigen Umgebung liefern beide Verfahren ähnliche Ergebnisse (vgl. Abbildung A.6). Da das Zählverfahren genauso gute Ergebnisse liefert, wie das Bayes-Verfahren, in einigen Fällen sogar bessere, ist das Zählverfahren dem Bayes-Verfahren in der momentanen Konfiguration vorzuziehen.



(a) Bayes-Verfahren



(b) Zählverfahren

Abbildung 5.9.: Vergleich der komplexen Map

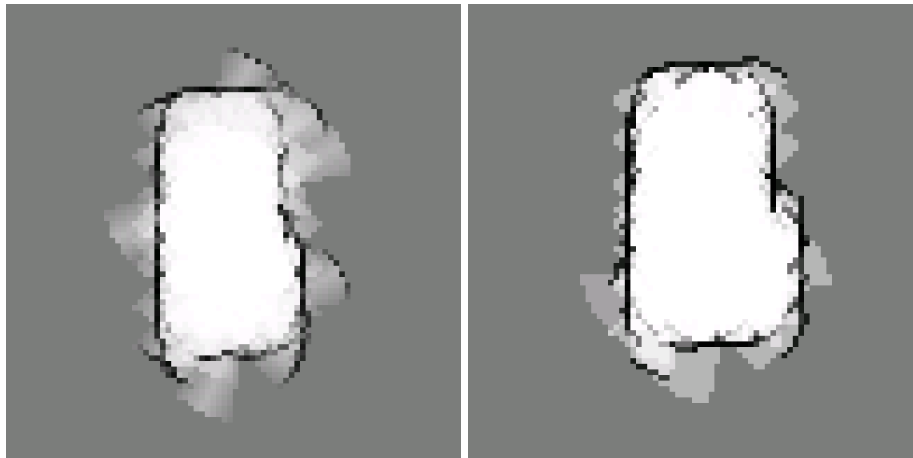
Ein Schwachstelle stellen Ecken in der Umgebung dar. Dies lässt sich, sowohl in der einfachen, wie auch in der komplexen Umgebung, in den erstellten Maps beobachten. Da weder mit Anpassung der Parameter, wie im Kapitel 5.2 beschrieben, noch durch verschiedene Integrationsverfahren dieses Problem reduziert werden konnte und auch durch [Müller, 2011] bekannt ist, dass die Ultraschallsensoren winkelabhängig sind, lässt sich dieses Problem auf die Ultraschallsensoren zurückführen.

5.4. Auswirkungen der Wahrscheinlichkeitsverteilung

Hier wurden die Auswirkungen der Wahrscheinlichkeitsverteilung (siehe Kapitel 3.3) auf das Mapping-Verfahren evaluiert. Dabei wurden für die Evaluierung mit Wahrscheinlichkeitsverteilung die Ergebnisse aus der Optimierung herangezogen und mit Ergebnissen ohne Verteilung verglichen. Zu beachten ist, dass die Ergebnisse ohne Verteilung den Sensorversatz gegenüber der Quadrocopterposition berücksichtigen, während die aus der Optimierung verwendeten Ergebnisse den Versatz vernachlässigen. Dies hat zur Folge, dass die erstellten Maps, ohne Verteilung, größer sind und die Abmessungen der eigentlichen Umgebung entsprechen. Dies hat jedoch keine weiteren Auswirkungen auf das Bayes- und das Zählverfahren.

5.4.1. Auswirkungen auf das Bayes-Verfahren

Mit Werten von $p(\mathbf{m}_i|z_t, x_t)_{free} < 0,4$ und $p(\mathbf{m}_i|z_t, x_t)_{occ} > 0,6$, ergaben sich im Bayes-Verfahren, mit und ohne Wahrscheinlichkeitsverteilung auf dem Sensorsektor, nur unzureichende Ergebnisse, wie bereits in Kapitel 5.2.4 beschrieben. Es treten auch diesselben Probleme auf. Mit trägeren Werten ergaben sich in beiden Varianten gute Maps. Dabei sind jedoch, wenn die Wahrscheinlichkeitsverteilung nicht genutzt wird, die erkannten Wände etwas besser definiert (vgl. Tabelle A.15). Für die optimalen Werte des Bayes-Verfahrens wird ohne Verteilung ein ebenfalls gutes Ergebnis erzielt. Es entstehen jedoch etwas mehr Artefakte innerhalb der Map, die auf die fehlende Wahrscheinlichkeitsverteilung und die damit nicht mehr vorhandene Abschwächung im Randbereich des Ultraschallsensors zurück zu führen sind. Die Nutzung einer Wahrscheinlichkeitsverteilung ist, in der aktuellen Konfiguration, durch die entstehenden Artefakte, einem Sensormodell ohne Verteilung vorzuziehen.



(a) Mit Wahrscheinlichkeitsverteilung (b) Ohne Wahrscheinlichkeitsverteilung

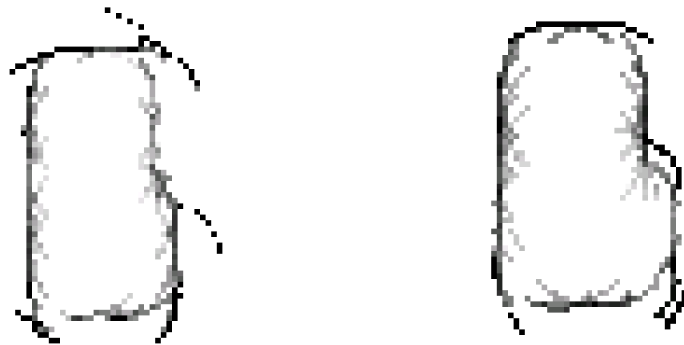
Abbildung 5.10.: Map aus Bayes-Verfahren mit $p(\mathbf{m}_i|z_t, x_t)_{free} = 0,4775$ und $p(\mathbf{m}_i|z_t, x_t)_{occ} = 0,55$

5.4.2. Auswirkungen auf das Zählverfahren

Durch die Anwendung der Wahrscheinlichkeitsverteilung auf das Zählverfahren, ergaben sich ähnliche Ergebnisse wie ohne Verteilung. Im Gegensatz zum Bayes-Verfahren (vgl. Kapitel 5.4.1) zeigt sich im Zählverfahren jedoch kaum eine Verbesserung der Wanderkennung, wie Tabelle A.16 entnommen werden kann. Lediglich die Artefakte nehmen ohne die Wahrscheinlichkeitsverteilung zu, was sich, wie beim Bayes-Verfahren, auf die fehlende Abschwächung im Randbereich durch die Verteilung zurück führen lässt. Mit den in Kapitel 5.2.5 angenommen optimalen Werten ergibt sich mit der Wahrscheinlichkeitsverteilung eine etwas bessere Map (siehe Abbildung 5.11).

5.5. Zusammenfassung

Im vorherigen Kapitel wurde die Evaluierung der implementierten Verfahren beschrieben und deren Ergebnisse analysiert. Dabei wurde ersichtlich, dass mit der vorhandenen Hard- und Software ein Mapping-Verfahren möglich ist. Trotzdem gibt es immer noch Probleme die durch die angewandten Mapping-Verfahren nicht lösbar waren. So kann das Sensormodell noch weiter verfeinert und an die verwendeten Ultraschallsensoren angepasst werden. Auch muss die Positionsbestimmung noch angepasst werden, da im realen Betrieb kein optisches Tracking System zur Verfügung steht und die Position über Ultraschallsensoren ungenau ist. Das Mapping wurde zudem an wenigen Positionen durchgeführt, was nur bedingt der Realität entspricht. Im Flug hingegen werden Messungen meist



(a) Mit Wahrscheinlichkeitsverteilung (b) Ohne Wahrscheinlichkeitsverteilung

Abbildung 5.11.: Map aus Zählverfahren mit $m_{free} = 1$ und $m_{occ} = 2$

an mehreren leicht unterschiedlichen Positionen durchgeführt. Insgesamt ist das System noch nicht flugfähig. Um das Mapping auf dem Quadrocopter auch im Flugbetrieb durchzuführen, sind noch Verbesserungen des Systems notwendig. Hierfür muss das Mapping, und eventuell auch die Hardware, weiter angepasst und erweitert werden.

6. Diskussion und Ausblick

6.1. Diskussion

Die Evaluierung hat gezeigt, dass die Ultraschallsensoren bis zu einer Reichweite von 150cm brauchbare Ergebnisse liefern. Die mögliche Reichweite von 600cm voll auszunutzen war mit den implementierten Sensormodell nicht möglich, da keine brauchbare Map erstellt werden konnte. Der große Öffnungswinkel des Sensors verursachte, wie die Evaluierung zeigt, zu viele Fehlinterpretationen von Zellen. Die zusätzliche Einschränkung des Sensoröffnungswinkel brachte keine Verbesserung der Ergebnisse. In Zusammenhang mit einer höheren Reichweite gewinnt dieser Parameter jedoch an Bedeutung. Ein eingeschränkter Öffnungswinkel bringt jedoch, genau wie das stark vereinfachte Strahlenmodell, den Nachteil, das sich Hindernisse, die in einem Winkel zum Sensor erkannt wurden, in die Mitte des Sensorkegels geschoben werden.

Die Anpassung der Zellengröße brachte die erwartete Änderung des Detailgrades, jedoch stieg die notwendige Rechenleistung mit jeder Verkleinerung der Zellen stark an. Dabei reichte bei kleinen Zellen selbst die Rechenleistung des PC bei einer Zellengröße von 2cm nicht mehr aus, um das Mapping zwischen den Sensormessungen durchzuführen.

Das Bayes-Verfahren funktioniert, ist aber besonders anfällig für lange Standzeiten an einer Position. Dabei werden in Zusammenhang mit dem Sektormodell häufig bereits als korrekt erkannte Zellen wieder als undefiniert oder frei beobachtet, wodurch sich Fehler in der Map bilden. Die Optimierung der Wahrscheinlichkeiten kann dieses Problem verringern, jedoch nicht vollständig lösen. Das Zählverfahren hingegen liefert leicht bessere Ergebnisse als der Bayes-Filter. Die Anpassung der Inkrementierungsvariablen hat sich als nur bedingt hilfreich beim Optimieren des Verfahrens erwiesen. Es ist aber weniger leistungsintensiv als das Bayes-Verfahren, da, anstelle von logarithmischen Berechnungen, lediglich Zähler inkrementiert werden müssen.

Besondere Probleme ergaben sich auf dem Controllerboard durch den geringen SRAM. Da die Map auf die SD-Karte ausgelagert wurde, nahm die Geschwindigkeit des Mappings stark ab. Selbst mit dem einfacheren Strahlenmodell konnte das Mapping nicht in annehmbarer Ausführungsgeschwindigkeit durchgeführt werden. Die SD-Karte spielt hierbei die größte Rolle, da die häufigen Schreib- und Lesezugriffe sehr langsam sind. Trotz eines Puffers, der die am häufigsten genutzten Knoten im Speicher des Microcontrollers hält, konnte das Geschwindigkeitsproblem nicht gelöst werden. Ein Mapping auf dem Microcontroller direkt, ist mit dem beschriebenen Algorithmus in Echtzeit nicht möglich.

Auch besitzen die Sensoren einen sehr großen Öffnungswinkel, weswegen es nicht selten zu fehlerhaften Integrationen in der Karte kommt. Besonders betroffen sind hiervon die Ecken von Räumen. Das Problem konnte zwar mit der Optimierung der Parameter eingedämmt werden, aber nicht vollständig gelöst. Das für die Ultraschallsensoren verwendete Sektorenmodell mit einer linearen Wahrscheinlichkeitsverteilung hilft geringfügig fehlerhafte Integrationen am Rande des Sensorkegels zu verringern.

Fazit In idealisierten Umgebungen und mit guten Messdaten funktioniert das Mapping. Die Evaluierung hat gezeigt, dass Hindernisse erkannt werden und auch korrekt in die Map integriert werden. Für den realen Betrieb ist es jedoch aufgrund fehlender bzw. schlechter Positionsdaten, aus den Ultraschallsensoren, nicht geeignet. Auch stellt die Leistungsanforderung noch ein Problem dar, welches für den autonomen Betrieb noch gelöst werden muss. Aus diesen Gründen ist es für die Anwendung im Quadrocopter noch nicht geeignet.

6.2. Ausblick

Durch die Evaluierung wurde gezeigt, dass das Mapping funktioniert. Für den Einsatz im realen Betrieb sind jedoch noch einige Verbesserungen notwendig. Die Positionsbestimmung spielt hierbei eine wichtige Rolle. In Verbindung mit einem optischen Fluss Sensor, der die Position bestimmt, würde das System deutlich verbessert und eventuelle flugfähig sein.

6.2.1. Fehlerhaft bzw. schlecht erkannte Zellen

Wie die Evaluierung gezeigt hat, traten immer wieder fehlerhafte Erkennung von Zellen auf, in denen eigentlich blockierte Zellen als frei oder freie Zellen als blockiert in die Map eingehen. Dies ist bedingt durch den breiten Sensorkegel der verwendeten Ultraschallsensoren. Dieser Effekt könnte verringert werden, indem die Nachbarzellen der vom Sensor beobachteten Zelle beachtet werden. Dadurch könnte eine sicherere Aussage über die vom Sensor erfasste Zelle getroffen werden. Ist eine als blockiert beobachtete Zelle von freien Zellen umgeben, ist die Wahrscheinlichkeit einer Fehlerkennung sehr hoch und die Messung könnte verworfen werden oder die Zelle wird nicht als blockiert in die Map integriert. Dies gilt ebenfalls für frei beobachtete Zellen, die von blockierten Zellen umgeben sind.

Auch wäre es möglich, Zellen, die einen gewissen Grenzwert über- oder unterschritten haben, und damit ausreichend gut als blockiert oder frei gelten, nicht zu ändern. Eine Messung, die eine ausreichend gut definierte Zelle einschließt, würde diese ignorieren.

Eine weitere Möglichkeit fehlerhafte Messungen einzugrenzen wäre den Quadrocopter nach jeder Messung um die Yaw-Achse zu drehen und weitere Messungen durchzuführen. Danach wird für jeden Sensor das Minimum aus den Messungen bestimmt. Dadurch werden Fehlerkennung durch zu flaches auftreten des Schalls auf das Hinderniss reduziert.

Auch das verwendete Sektormodell könnte, mit einer besser an die Ultraschallsensoren angepassten Wahrscheinlichkeitsverteilung, helfen, fehlerhafte Integrationen von Zellen zu verringern. Eine Gauß-Verteilung oder eine Verteilung, die an die Signalausbreitung des Sensors angelehnt ist, könnte das Sektormodell weiter verbessern.

Eine Sensorfusion der momentan sich überschneidenden Ultraschallsensoren kann den effektiven Öffnungswinkel der Sensoren reduzieren. Dabei werden die zwei sich überlagernden Sensormessungen miteinander verglichen. Liefern beide Sensoren innerhalb eines Toleranzbereichs gleiche Werte, so ist es wahrscheinlich, dass das Hindernis im Überschneidungsbereich der Sensoren liegt. Sind die beiden Werte nicht gleich, wurden wahrscheinlich zwei Hindernisse in den sich nicht überschneidenden Bereichen erfasst.

Der Einsatz weiterer Sensorik im Zusammenhang mit einer Sensorfusion kann diese Problem noch weiter reduzieren. Dabei könnten Infrarotsensoren am Quadrocopter ausgerichtet werden, sodass die Infrarotsensoren den Öffnungsbereich der Ultraschallsensoren in kleinere Bereiche unterteilen. Durch geschickte Sensorfusion könnte somit der effektive Öffnungswinkel der Ul-

trasschallsensoren verkleinert werden. Durch einen Vergleich der Ultraschallmessung mit den Infrarotsensoren im Öffnungsbereich des Ultraschallsensors, kann der Bereich des Ultraschallsensors in dem sich das Hindernis befindet, bestimmt werden. Fehlerhafte Integrationen würden somit weiter verringert.

Mit Hilfe weiterer Sensorik, die weniger winkelabhängig ist als Ultraschallsensoren, könnte die Erkennung von Ecken in der Umgebung ebenfalls verbessert werden.

6.2.2. Leistungsprobleme

Um die Leistungsprobleme des Mappings zu verringern oder gar zu lösen, muss die Software noch weiter optimiert werden. So wird der Quad-Tree für jede Zelle, die erfasst wurde, einmal rekursiv nach unten und danach wieder nach oben traversiert, um die Elternelemente zu aktualisieren. Da die Sensoren Zellen erfassen, die sich meist in unmittelbarer Nachbarschaft befinden, besitzen Zellen häufig dasselbe Elternelement. Um ein unnötiges Durchlaufen des Baumes zu verringern, könnte zuerst die unterste Ebene des Baumes aktualisiert werden und danach aufsteigend die darüber liegenden Ebenen. Für eine Messung mit n erfassten Zellen, muss der Baum dadurch nicht mehr $2n$ mal durchlaufen werden, sondern nur noch $n + 1$ mal. Für jede Zelle wird der Baum dabei einmal nach unten durchlaufen, um die Zelle anzulegen, falls diese nicht existiert und diese zu aktualisieren, sowie einmal nach oben um die Elternelemente zu aktualisieren.

Um neue Zellen nicht im Betrieb im Baum anlegen zu müssen, könnte der gesamte Quad-Tree bei der Initialisierung einmalig angelegt werden. Dies kann sich jedoch nachteilig auswirken wenn die Karte über die zu Beginn festgelegten Abmessungen hinaus geht und ein neuer Teilbaum, mit einem neuen gemeinsamen Wurzelement, angelegt werden muss.

Auf dem Controllerboard stellt die SD-Karte einen Flaschenhals dar. Da der verwendete Microcontroller über zwei DMA-Controller¹ (Direct Memory Access) verfügt, könnte die Nutzung des DMA für den SPI-Bus an dem die SD-Karte angeschlossen ist, für eine deutlich höhere Geschwindigkeit beim Zugriff auf die SD-Karte sorgen. Auch der Zugriff auf die SD-Karte kann durch die Nutzung von größeren Blöcken beim Schreiben und Lesen eventuell beschleunigt werden.

¹In [at32uc3, 2012] auch als PDCA bzw. PDCB bezeichnet.

Literaturverzeichnis

[atmel 2007] AT32UC3 Software Framework, 2007

[at32uc3 2012] 32-bit Atmel AVR Microcontroller AT32UC3A0512, 2012

[wiki 2012] : Wikipedia. 2012. – URL <http://de.wikipedia.org>

[Bošnak und Blažič 2012] BOŠNAK, Matevž ; BLAŽIČ, Sašo: Sparse VSLAM with camera-equipped quadrocopter. In: *Proceedings of the Third international conference on Autonomous and Intelligent Systems*. Berlin, Heidelberg : Springer-Verlag, 2012 (AIS'12), S. 135–140

[Cormen et al. 2009] CORMEN, T.H. ; LEISERSON, C.E. ; RIVEST, R.L. ; STEIN, C.: *Introduction to Algorithms*. MIT Press, 2009

[Haehnel 2004] HAEHNEL, Dirk: *Mapping with Mobile Robots*, Albert-Ludwigs-Universität Freiburg im Breisgau, Dissertation, 2004

[Klawonn 2012] KLAWONN, F.: *Introduction to Computer Graphics*. Springerverlag London Limited, 2012 (Undergraduate Topics in Computer Science)

[Krause 2012] KRAUSE, Stefan: Multi-Purpose Environment Awareness Approach for Single Line Laser Scanner in a Small Rotorcraft UA. In: *Journal of Intelligent and Robotic Systems* 65 (2012), Nr. 1-4, S. 587–601

[Kuipers 2000] KUIPERS, J. B.: Quaternions and Rotation Sequences. In: *Geometry, Integrability and Quantization* 1 (2000), S. 127–143

[Müller 2011] MÜLLER, Thilo: *Implementierung und Evaluierung eines Systems zur Hinderniserkennung und Kollisionsvermeidung für Indoor-Quadrocopter*, Julius-Maximilians-Universität Würzburg, Diplomarbeit, 2011

-
- [Thrun 1998] THRUN, S.: Learning metric-topological maps for indoor mobile robot navigation.
In: *Artificial Intelligence* 99 (1998), S. 21–71
- [Thrun et al. 2005] THRUN, S. ; BURGARD, W. ; FOX, D.: *Probabilistic Robotics*. Mit Press,
2005 (Intelligent Robotics and Autonomous Agents)

A. Anhang

A.1. Messdaten umrechnen

A.1.1. Messdaten in Quadrocoptersystem umwandeln

Zuerst wird der gemessene Wert, welcher als u bezeichnet wird, in das Quadrocoptersystem umgerechnet. Dazu werden die Ausrichtungen der Sensoren am Quadrocopter um die z -Achse, sowie um die x, y -Ebene benötigt, welche als α beziehungsweise β bezeichnet werden. α bewegt sich hierbei im Intervall $[-\pi, \pi]$ und β im Intervall $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Weiterhin wird der Versatz \vec{x}_v des Sensors zum Ursprung des Quadrocoptersystems, dem Masseschwerpunkt aufaddiert. Zur Umrechnung in das Quadrocoptersystem \vec{x}_q gelten hierbei folgende Gleichungen.

$$\vec{x}_q = \begin{pmatrix} \cos \alpha \cdot \cos \beta \cdot u \\ \sin \alpha \cdot \cos \beta \cdot u \\ -\sin \beta \cdot u \end{pmatrix} + \begin{pmatrix} x_v \\ y_v \\ z_v \end{pmatrix} \quad (\text{A.1})$$

Da der Versatz und die Ausrichtung des Sensors am Quadrocopter durch den mechanischen Aufbau bestimmt werden, kann vorausgesetzt werden dass diese bekannt sind.

A.1.2. Umwandlung in Weltkoordinaten

Nachdem die Messung nun im Quadrocoptersystem vorliegt, wird diese in das Weltkoordinatensystem gedreht. Dazu wird \vec{x}_q (Gleichung A.1) nach Gleichung A.2 in ein Quaternion q_u umgerechnet.

$$q_u = \begin{pmatrix} 0 \\ \vec{x}_q \end{pmatrix} \quad (\text{A.2})$$

Nun wird q_u mit Hilfe des Ausrichtungsquaternions des Quadrocopters q_Q gedreht.

$$q = q_Q \cdot q_u \cdot q_Q^{-1} \quad (\text{A.3})$$

Das Inverse des Quaternions q_Q^{-1} aus Gleichung A.3 berechnet sich, unter der Voraussetzung das q_Q bereits ein Einheitsquaternion mit $|q_Q| = 1$ ist, wie folgt. [Kuipers, 2000]

$$q_Q^{-1} = \begin{pmatrix} q_{Q0} \\ -iq_{Q1} \\ -jq_{Q2} \\ -kq_{Q3} \end{pmatrix} \quad (\text{A.4})$$

A.2. Bresenham Sektoreinteilung

Da der Bresenham-Algorithmus lediglich mit positive Steigungen bis 1 anwendbar ist, wurde die Anfangsbedingung D_{init} und die Inkrementierungsvariable Δ nach der Herleitung aus [Klawonn, 2012] für andere Steigungswerte nachgerechnet und wie folgt angepasst (siehe Tabelle A.1). Der Algorithmus wurde dabei in acht Bereiche unterteilt, wie in Abbildung A.1 dargestellt, in denen unterschiedliche Anfangs- und Inkrementierungsbedingungen gelten.

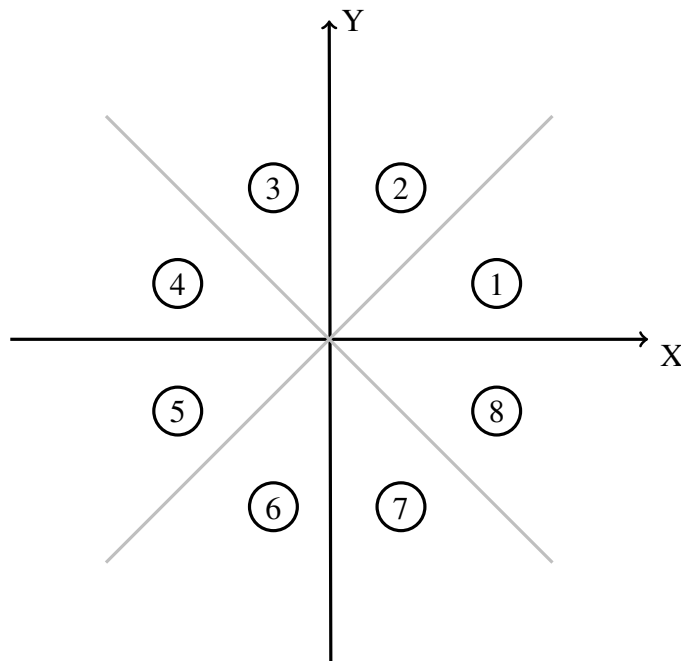


Abbildung A.1.: Einteilung der Sektoren für den Bresenham Algorithmus

Sektor	D_{init}	D_{new}
1	$2 \cdot dy - dx$	$D_{old} + \Delta$ mit $\begin{cases} 2 \cdot dy & \text{für } D_{old} \leq 0 \\ 2 \cdot (dy - dx) & \text{für } D_{old} > 0 \end{cases}$
2	$2 \cdot dx - dy$	$D_{old} + \Delta$ mit $\begin{cases} 2 \cdot dx & \text{für } D_{old} \leq 0 \\ 2 \cdot (dx - dy) & \text{für } D_{old} > 0 \end{cases}$
3	$-2 \cdot dx - dy$	$D_{old} + \Delta$ mit $\begin{cases} -2 \cdot dx & \text{für } D_{old} \leq 0 \\ -2 \cdot (dx + dy) & \text{für } D_{old} > 0 \end{cases}$
4	$-2 \cdot dy - dx$	$D_{old} + \Delta$ mit $\begin{cases} -2 \cdot dy & \text{für } D_{old} \leq 0 \\ -2 \cdot (dy + dx) & \text{für } D_{old} > 0 \end{cases}$
5	$-2 \cdot dy + dx$	$D_{old} + \Delta$ mit $\begin{cases} -2 \cdot dy & \text{für } D_{old} \leq 0 \\ 2 \cdot (dx - dy) & \text{für } D_{old} > 0 \end{cases}$
6	$-2 \cdot dx + dy$	$D_{old} + \Delta$ mit $\begin{cases} -2 \cdot dx & \text{für } D_{old} \leq 0 \\ 2 \cdot (dy - dx) & \text{für } D_{old} > 0 \end{cases}$
7	$2 \cdot dx + dy$	$D_{old} + \Delta$ mit $\begin{cases} 2 \cdot dx & \text{für } D_{old} \leq 0 \\ 2 \cdot (dx + dy) & \text{für } D_{old} > 0 \end{cases}$
8	$2 \cdot dy + dx$	$D_{old} + \Delta$ mit $\begin{cases} 2 \cdot dy & \text{für } D_{old} \leq 0 \\ 2 \cdot (dy + dx) & \text{für } D_{old} > 0 \end{cases}$

Tabelle A.1.: Entscheidungsvariable und Anfangsbedingung des Bresenham-Algorithmus

A.3. Optimierung der Mapping Parameter

A.3.1. Verwendete Parameter

In den Tabellen zu den Mapping Parametern beschreibt $p(\mathbf{m}_i|z_t, x_t)_{free}$ die Bayes-Wahrscheinlichkeit für freie Zellen und $p(\mathbf{m}_i|z_t, x_t)_{occ}$ die Wahrscheinlichkeit für blockierte Zellen wie in Gleichung 3.2 und 3.3 verwendet. m_{free} und m_{occ} beschreiben die Inkrementierungsvariablen für das Zählverfahren aus Gleichung 3.4. Die Anzahl der Messungen gibt an wie viele Messungen *pro Position* durchgeführt und integriert wurden. Dabei wurden für jeweils eine Messung alle 12 Ultraschallsensoren verwendet.

Parameter	$p(\mathbf{m}_i z_t, x_t)_{free}$	$p(\mathbf{m}_i z_t, x_t)_{occ}$	Öffnungswinkel	Zellengröße
Wert	0,525	0,475	55°	8cm
Parameter	m_{free}	m_{occ}	Anzahl Messungen	Sensorreichweite
Wert	2	2	10	siehe Tabelle A.7

Tabelle A.2.: Mapping Parameter Sensorreichweite

Parameter	$p(\mathbf{m}_i z_t, x_t)_{free}$	$p(\mathbf{m}_i z_t, x_t)_{occ}$	Öffnungswinkel	Zellengröße
Wert	0,525	0,475	siehe Tabelle A.8	8cm
Parameter	m_{free}	m_{occ}	Anzahl Messungen	Sensorreichweite
Wert	2	2	10	150cm

Tabelle A.3.: Mapping Parameter der Öffnungswinkelanpassung

Parameter	$p(\mathbf{m}_i z_t, x_t)_{free}$	$p(\mathbf{m}_i z_t, x_t)_{occ}$	Öffnungswinkel	Zellengröße
Wert	0,525	0,475	35°	siehe Tabelle A.8
Parameter	m_{free}	m_{occ}	Anzahl Messungen	Sensorreichweite
Wert	2	2	10	150cm

Tabelle A.4.: Mapping Parameter der Anpassung der Zellengröße

Parameter	$p(\mathbf{m}_i z_t, x_t)_{free}$	$p(\mathbf{m}_i z_t, x_t)_{occ}$	Öffnungswinkel	Zellengröße
Wert	siehe Tabelle A.10		35°	8cm
Parameter	m_{free}	m_{occ}	Anzahl Messungen	Sensorreichweite
Wert	2	2	10	150cm

Tabelle A.5.: Mapping Parameter der Optimierung der Bayes-Wahrscheinlichkeiten

Parameter	$p(\mathbf{m}_i z_t, x_t)_{free}$	$p(\mathbf{m}_i z_t, x_t)_{occ}$	Öffnungswinkel	Zellengröße
Wert	0,4775	0,55	35°	8cm
Parameter	m_{free}	m_{occ}	Anzahl Messungen	Sensorreichweite
Wert	siehe Tabelle A.11		10	150cm

Tabelle A.6.: Mapping Parameter der Optimierung der Inkrementierungsvariablen

A.3.2. Ergebnisse der Optimierung

Die Maps werden als Graustufen Bilder dargestellt. Je dunkler eine Zelle dargestellt ist, desto wahrscheinlicher ist sie blockiert. Genauso gilt, je heller eine Zelle dargestellt ist, desto wahrscheinlicher ist sie frei. Der Verlauf der Wahrscheinlichkeit ist in Abbildung A.2 dargestellt.

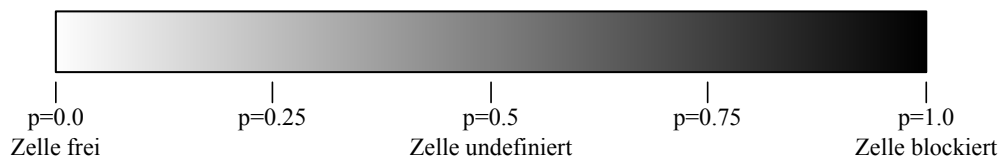


Abbildung A.2.: Wahrscheinlichkeit als Graustufen

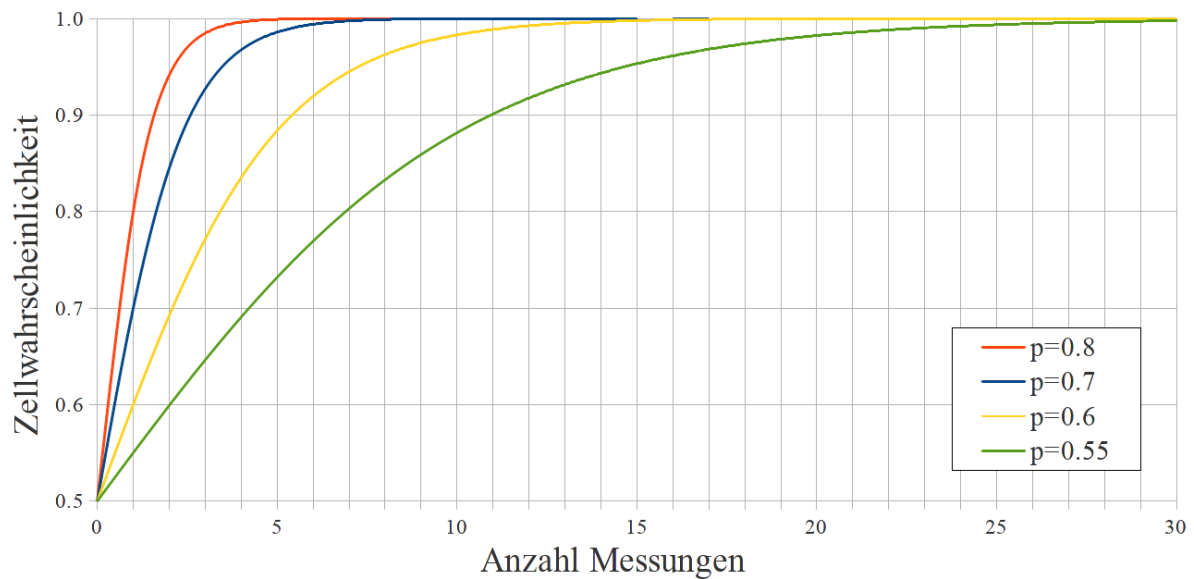


Abbildung A.3.: Wahrscheinlichkeitsverlauf einer als blockiert gemessenen Zelle mit $p(\mathbf{m}_i|z_t, x_t)_{occ} = p$

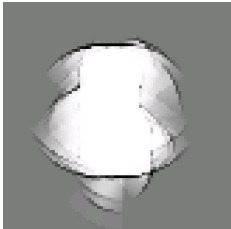

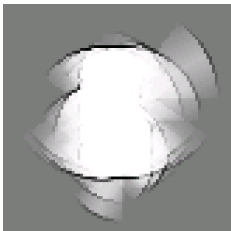

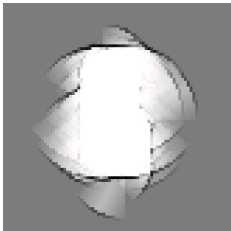

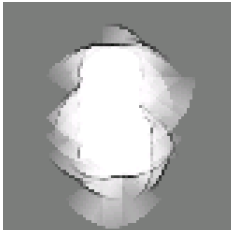

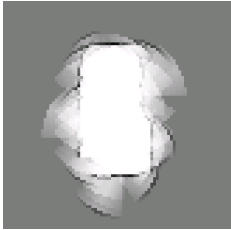

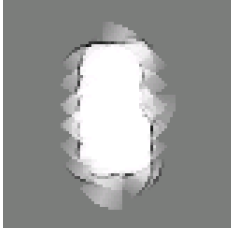

Sensorreichweite	Bayes-Verfahren	Zählverfahren
600cm		
300cm		
250cm		
200cm		
175cm		
150cm		

Tabelle A.7.: Ergebnis der Sensorreichweitenanpassung

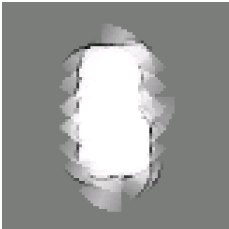

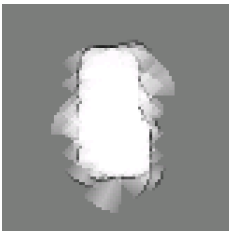

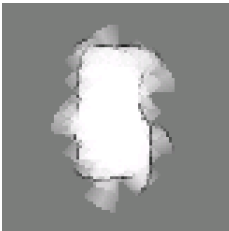

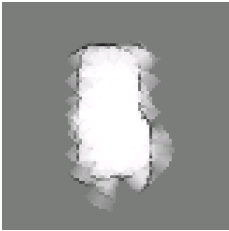



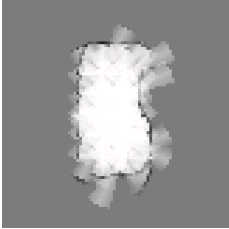

Öffnungswinkel	Bayes-Verfahren	Zählverfahren
55°		
45°		
40°		
35°		
30°		
27,5°		

Tabelle A.8.: Ergebnis der Öffnungswinkelanpassung



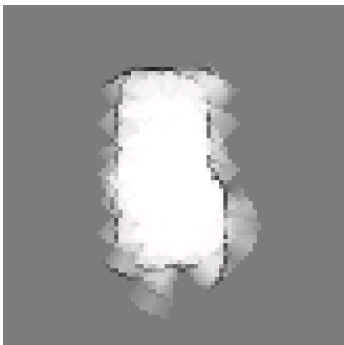



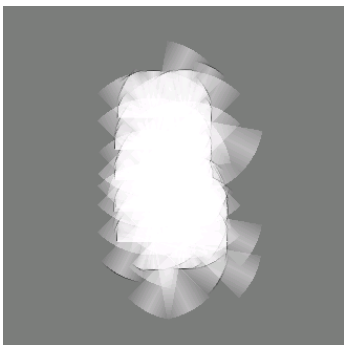

Zellengröße	Bayes-Verfahren	Zählverfahren
16cm		
8cm		
4cm		
2cm		

Tabelle A.9.: Ergebnis der Anpassung der Zellengröße

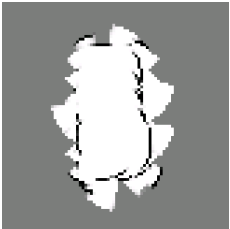
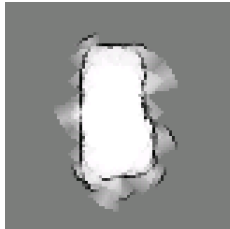
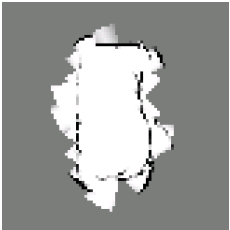
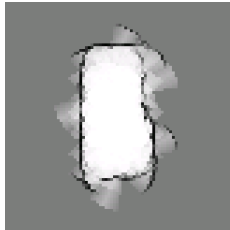
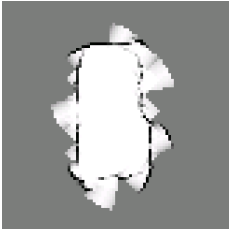
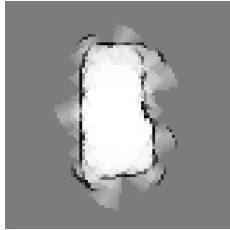
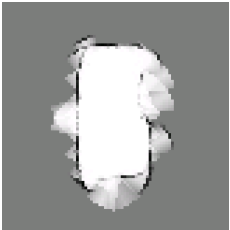
$p(\mathbf{m}_i z_t, x_t)$			$p(\mathbf{m}_i z_t, x_t)$		
p_{free}	p_{occ}	Map	p_{free}	p_{occ}	Map
0,2000	0,8000		0,4750	0,5500	
0,3000	0,7000		0,4775	0,5500	
0,4000	0,6000		0,4780	0,5500	
0,4500	0,5500				

Tabelle A.10.: Ergebnis der Optimierung der Bayes-Wahrscheinlichkeiten










m_{free}	m_{occ}	$\frac{m_{occ}}{m_{free}}$	Map	m_{free}	m_{occ}	$\frac{m_{occ}}{m_{free}}$	Map
1	2	2		1	1	1	
2	5	2,5		5	4	0,8	
4	10	2,5		5	3	0,6	
1	3	3		2	1	0,5	
1	4	4					

Tabelle A.11.: Ergebnis der Optimierung der Inkrementierungsvariablen

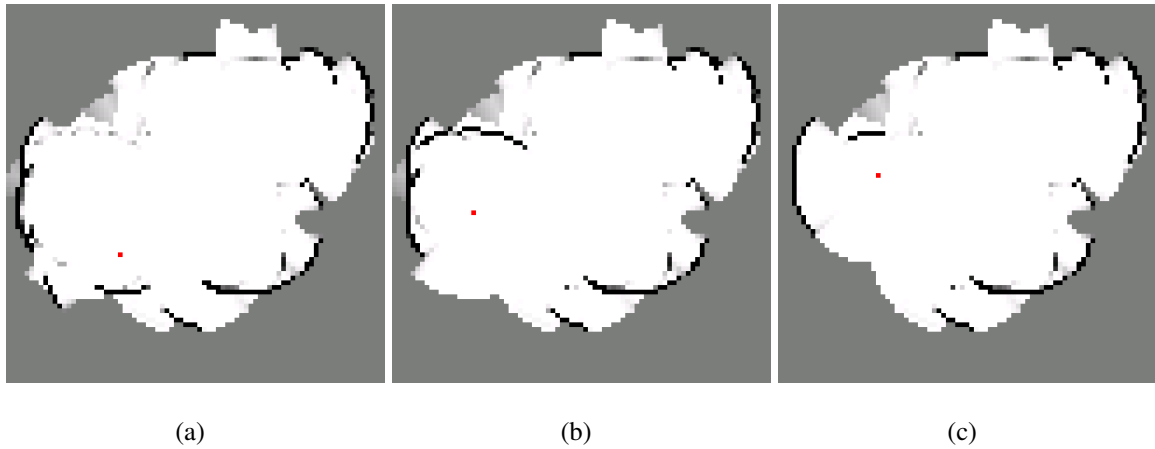


Abbildung A.4.: Komplexe Map erstellt mit Bayes-Verfahren¹

A.4. Ergebnis des Vergleichs der Mapping-Verfahren

Abbildung A.5 zeigt die komplexe Umgebung in der der Quadrocopter bewegt wurde. Messpositionen konnten nicht eingezeichnet werden, da diese nicht im Voraus festgelegt waren und zwischen den Messungen unterschiedlich wäre.

Parameter	$p(\mathbf{m}_i z_t, x_t)_{free}$	$p(\mathbf{m}_i z_t, x_t)_{occ}$	Öffnungswinkel	Zellengröße
Wert	0,4775	0,55	35°	4cm
Parameter	m_{free}	m_{occ}	Anzahl Messungen	Sensorreichweite
Wert	1	2	10	150cm

Tabelle A.12.: Mapping Parameter zum Vergleich zwischen Bayes- und Zählverfahren

¹Position rot gekennzeichnet

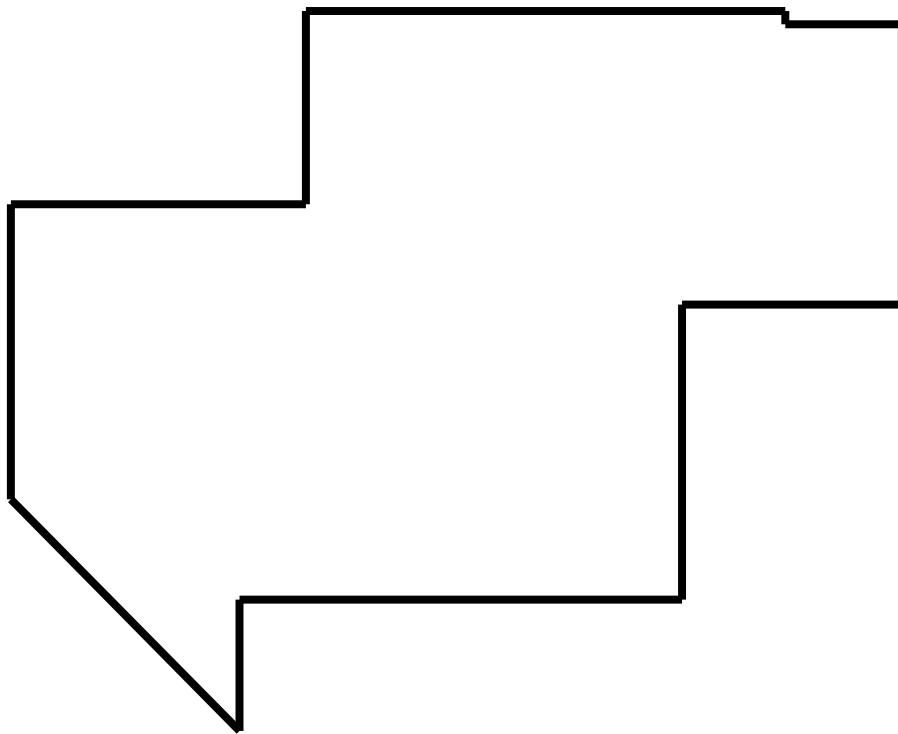
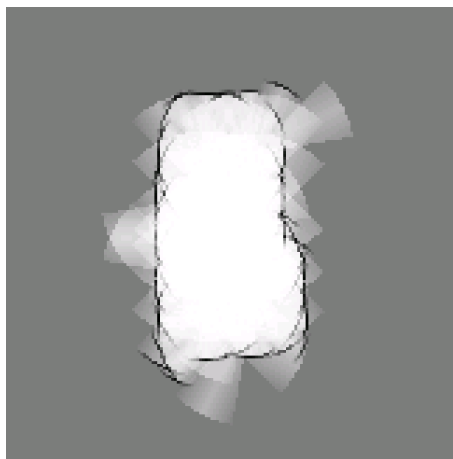


Abbildung A.5.: Komplexe Umgebung



(a) Bayes-Verfahren



(b) Zählverfahren

Abbildung A.6.: Vergleich der einfachen Map

A.5. Aufbau einer Map

A.5.1. Bayes-Verfahren

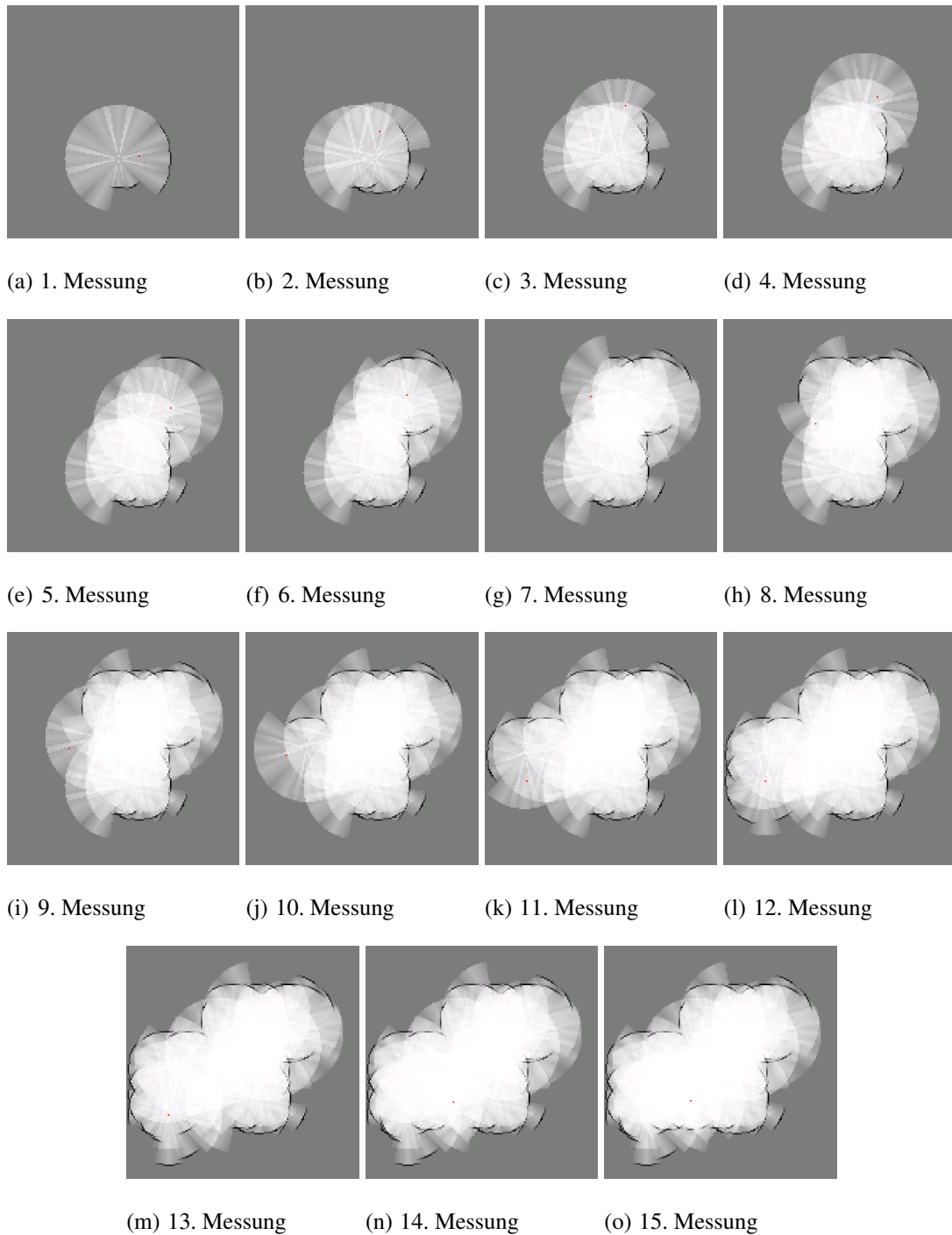


Abbildung A.7.: Aufbau einer Map mit dem Bayes-Verfahren²

A.5.2. Zählverfahren

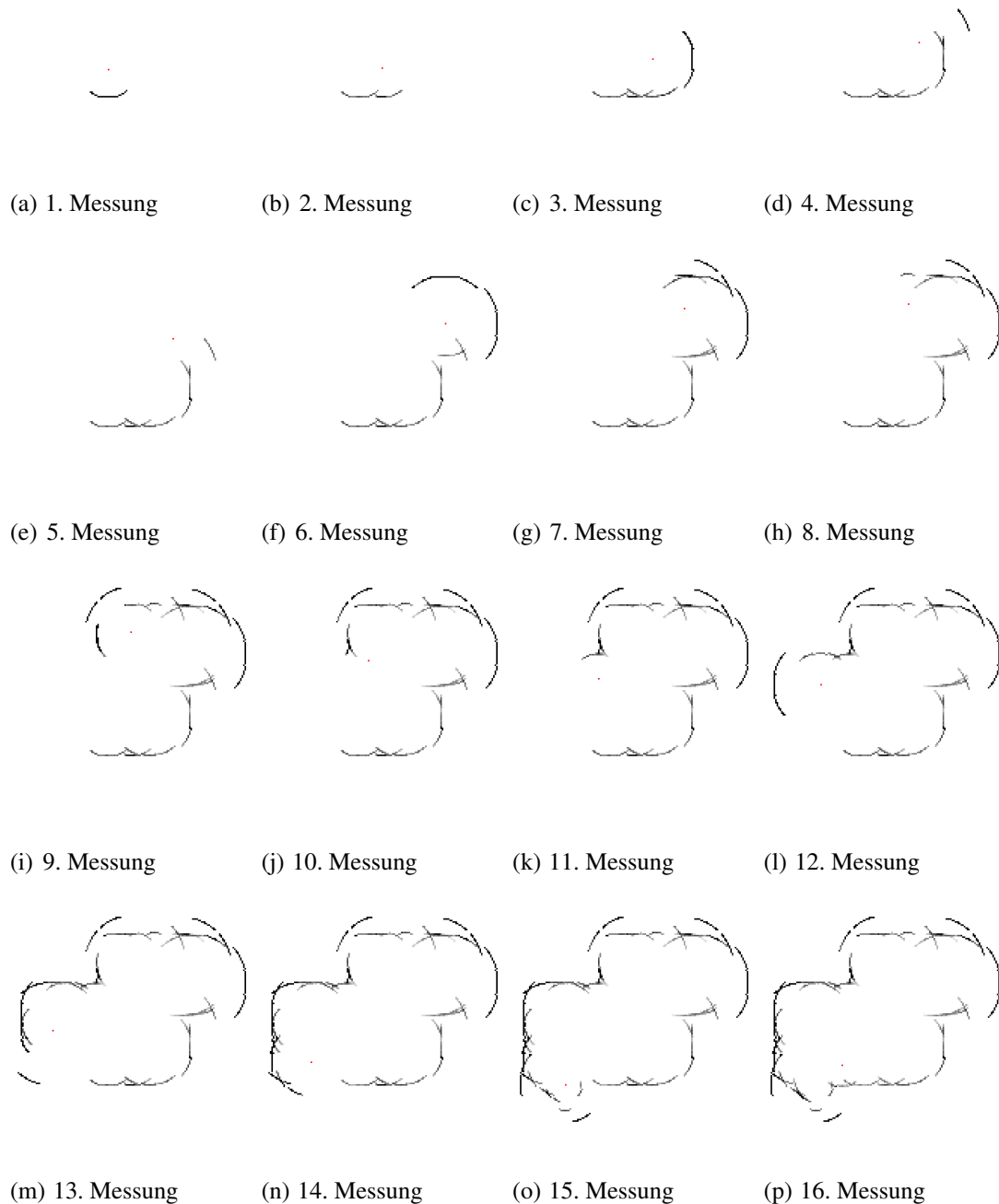


Abbildung A.8.: Aufbau einer Map mit dem Zählverfahren

²Die rot eingezeichneten Positionen stimmen nicht, da der Quadrocopter während des exportierens der Bitmaps bewegt wurde.

A.6. Auswirkung der Wahrscheinlichkeitsverteilung

Bei der Evaluierung mit der Wahrscheinlichkeitsverteilung auf dem Sensorsektor wurde der Versatz des Sensors, gegenüber der Position des Quadrocopters, nicht berücksichtigt. Bei der Evaluierung ohne Wahrscheinlichkeitsverteilung war der Versatz jedoch mitgerechnet. Aus diesem Grund ergibt sich für die gleiche Umgebung, ohne Verteilung, eine etwas größere Map.

A.6.1. Verwendete Parameter

Die folgenden Parameter wurden zur Evaluierung der Wahrscheinlichkeitsverteilung auf dem Sensorsektor verwendet. Die einzelnen Werten sind im Kapitel A.3.1 beschrieben.

Parameter Wert	$p(\mathbf{m}_i z_t, x_t)_{free}$ <i>siehe Tabelle A.10</i>	$p(\mathbf{m}_i z_t, x_t)_{occ}$	Öffnungswinkel 35°	Zellengröße 8cm
Parameter Wert	m_{free} 2	m_{occ} 2	Anzahl Messungen 10	Sensorreichweite 150cm

Tabelle A.13.: Mapping Parameter zur Evaluierung der Wahrscheinlichkeitsverteilung (Bayes-Verfahren)

Parameter Wert	$p(\mathbf{m}_i z_t, x_t)_{free}$ 0,4775	$p(\mathbf{m}_i z_t, x_t)_{occ}$ 0,55	Öffnungswinkel 35°	Zellengröße 8cm
Parameter Wert	m_{free} <i>siehe Tabelle A.11</i>	m_{occ}	Anzahl Messungen 10	Sensorreichweite 150cm

Tabelle A.14.: Mapping Parameter zur Evaluierung der Wahrscheinlichkeitsverteilung (Zählverfahren)

A.6.2. Bayes-Verfahren

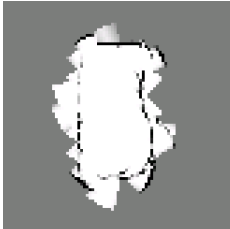

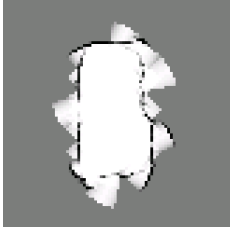





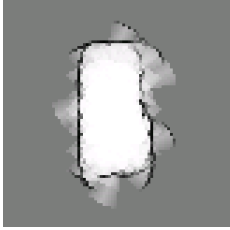
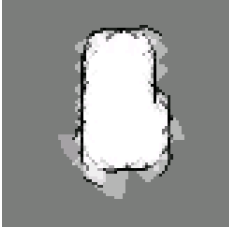
$p(\mathbf{m}_i z_t, x_t)$		Map	
p_{free}	p_{occ}	mit Verteilung	ohne Verteilung
0,3000	0,7000		
0,4000	0,6000		
0,4500	0,5500		
0,4750	0,5500		
0,4775	0,5500		

Tabelle A.15.: Auswirkungen der Wahrscheinlichkeitsverteilung auf das Bayes-Verfahren

A.6.3. Zählverfahren













m_{free}	m_{occ}	$\frac{m_{occ}}{m_{free}}$	Map	
			mit Verteilung	ohne Verteilung
1	1	1		
1	2	2		
2	5	2,5		
1	3	3		
5	4	0,8		
2	1	0,5		

Tabelle A.16.: Auswirkungen der Wahrscheinlichkeitsverteilung auf das Zählverfahren