

Untersuchungen der Einsatzmöglichkeiten mobiler Roboterplattformen zur Unterstützung von Videoüberwachungssystemen

Jakob Unger, Daniel Muhle

Institut für Photogrammetrie und GeoInformation
Leibniz Universität Hannover
Nienburger Straße 1, 30167 Hannover
eMail: unger@ipi.uni-hannover.de
URL: <http://www.ipi.uni-hannover.de/>

Zusammenfassung. Es wird ein Projekt vorgestellt, in dem Kamera und Roboter kombiniert werden. Der Roboter nimmt mittels verschiedener Sensoren die Umgebung auf und erstellt aus den gesammelten Informationen eine Karte. Er bewegt sich im Sichtbereich einer Überwachungskamera, wird anhand einer Signalisierung in deren Video verfolgt und es wird seine 3D-Position im Raum photogrammetrisch ermittelt. Zusätzlich bestimmt der Roboter seine Position per Odometrie selbst, so dass diese redundant vorliegt. Die Karte muss aus fehlerbehafteten Beobachtungen zusammengesetzt werden, was über einen häufigkeits-basierten Ansatz realisiert wird. Aufgrund des Einsatzes von Low Cost Sensoren ist mit geringen Genauigkeiten und mehrdeutigen Messungen umzugehen. Die so gewonnenen Informationen über die beobachtete Szene lassen sich als zusätzliche Information (Geometrie der Szene und Verdeckungen) für die automatische Auswertung des Videos verwenden.

1 Einleitung

In der Videoüberwachung müssen Aufgaben gelöst werden, die derzeit noch manuelles Eingreifen und den Einsatz von Expertenwissen erfordern. Dazu gehören Herausforderungen wie die Kalibrierung zoombarer Überwachungskameras, die Verknüpfung der Aufnahmen mehrerer Kameras in einem gemeinsamen Weltkoordinatensystem sowie die automatische Auswertung des Bildinhalts. Zur Unterstützung der Lösung bietet sich der Einsatz einer mobilen Roboterplattform an. Im Bereich der Objektverfolgung sind neben der Verdeckung von Objekten, ein möglicher Informationsverlust durch die 2D-Abbildung einer 3D-Szene und Echtzeitanforderungen an die Prozessierung, Punkte, die mithilfe eines Roboters unterstützt werden könnten. Ein Roboter kann sich im Sichtbereich der Kamera bewegen, mittels Sensoren die Umgebung erfassen oder sogar mit ihr interagieren. Nach Verlassen des Sichtbereichs der Kamera, können außerdem über diesen hinaus Daten gesammelt werden. Eine Kartierung der Umwelt durch einen Roboter kann beispielsweise für eine Objekterkennung im Rahmen des Personentrackings hilfreiche Informationen liefern und die Erstellung von Tiefenkarten für Kamerabilder unterstützen. Eine solche Information könnte die Ausrichtung und Lage des Fußbodens relativ zu einer oder mehreren Kameras oder die Vorhersage von Verdeckungen verfolgter Objekte im Kamerabild mithilfe von Verdeckungskarten sein. Liegt eine durch einen Roboter erfasste Karte der sichtbaren Umgebung vor, kann diese

für die zuvor genannten Aufgaben als Eingangsdatensatz verwendet werden. Trägt ein Roboter durch eine geeignete Signalisierung Objektpunkte bei, kann er auch die Kamerakalibrierung unterstützen.

Für verschiedenste Szenarien erscheint die Nutzung von Überwachungskameras relevant: Sie sind an den verschiedensten Orten montiert und in bestimmten Anlagen sogar vorgeschrieben. Damit stellen sie möglicherweise auch eine bereits vorhandene, spontan nutzbare Infrastruktur für den Einsatz von Robotern dar (Stichwort „Rettungsrobotik“). Dabei kann eine solche Kamera den Roboter oder umgekehrt der Roboter selbst die Auswertung der Kameravideos, gemäß der genannten Herausforderungen, unterstützen.

Im Folgenden wird ein Projekt vorgestellt, das das Zusammenspiel aus Kamera und Roboterplattform zur Kartierung der Umgebung nutzt.

2 Grundlagen

Als Kamera kommt die Netzwerk-Kamera AXIS 214 PTZ zum Einsatz. Sie besitzt die Fähigkeit den Sichtbereich durch Schwenken (pan), Neigen (tilt) und Zoomen zu variieren. Die Programmierung der Auswertung des gelieferten MJPG-Bild-Stroms erfolgt in C++ unter Nutzung der offenen Computer Vision Bibliothek OpenCV.

Die Roboterplattform ist eine Eigenentwicklung basierend auf einem LEGO Mindstorms NXT 2.0 Roboterbaukasten. Den Kern des Roboters bildet der per Bluetooth von einem PC ansprechbare sogenannte NXT-Baustein. Er verbindet einen Tast- und einen Ultraschallsensor, sowie zwei Servomotoren eines Gummiketten-Differentialantriebs mit der Steuerungslogik. Mittels eigenen Speichers und Prozessors kann der Roboter autonom agieren. Die Programmierung erfolgt über die alternative Firmware LeJOS in der Programmiersprache Java (Berns & Schmidt, 2010). Auf dem Roboter ist ein planares Schachbrettmuster angebracht (siehe Abbildung 2).



Abbildung 1: Messanordnung (ohne PC)

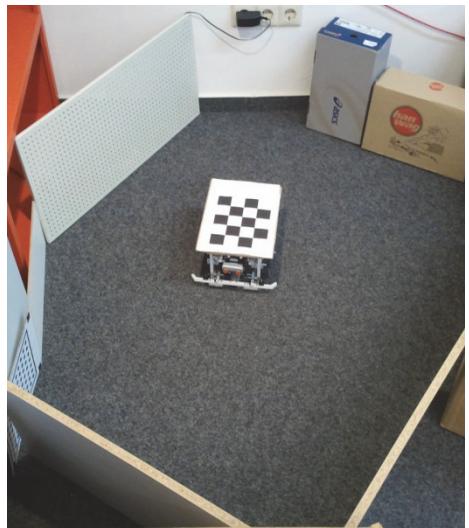


Abbildung 2: Testumgebung

Abbildung 1 zeigt die in etwa 2,50 Meter Höhe fast senkrecht nach unten auf eine Testumgebung blickende Kamera. In Abbildung 2 ist diese Testumgebung mit dem Roboter zu sehen. Die Kamera wird zunächst ohne Nutzung der PTZ-Funktionalität fest konfiguriert und kalibriert (Luhmann, 2010).

3 Methode

Fahrstrategie

Der an einer beliebigen Position in der Testumgebung platzierte Roboter soll diese mit möglichst kurzen Wegen und möglichst detailliert aufnehmen. Die Ausgangssituation ist die, dass keinerlei Informationen über die Umwelt existieren. Damit ist eine Planung des zu fahrenden Weges nicht möglich und die Fahrstrategie ist zunächst darauf beschränkt, die unbekannte Umgebung zu erfassen. Dies geschieht über eine Stoßstange als Tastsensor und einen nach vorne ausgerichteten Ultraschallsensor als Distanzmesser. Der Roboter muss sich fortbewegen, um die Umgebung zu erkunden und nach Möglichkeit alle Bereiche mittels Ultraschall anzuzielen. Die einfachste Strategie ist dabei, ihn geradeaus fahren zu lassen und die Richtung zu ändern, wenn die Kollision mit einem Hindernis droht. Dies ist der Fall, wenn die gemessene Distanz einen vorzugebenden Wert unterschreitet. Die Ultraschallmessung muss während der Geradeausfahrt also kontinuierlich erfolgen, was mithilfe eines entsprechenden Messmodus umgesetzt wird. Versagt der Ultraschallsensor bei der Erkennung eines Hindernisses und es kommt zu einer Kollision, detektiert dies der Tastsensor an der Stoßstange. In diesem Fall muss der Roboter vor der Richtungsänderung zurücksetzen, um sich drehen zu können. Wird nach der Drehung wieder ein Hindernis erkannt, wird die Geradeausfahrt nicht ausgeführt und wieder ausgewichen. Die Drehung erfolgt per Zufall nach rechts oder links um einen zufälligen Winkel in einem festzulegenden Gradbereich. Der Zufall für die Drehrichtung wird jedoch eingeschränkt: Um ein länger andauerndes Drehen des Roboters in Ecken zu vermeiden, wird die Richtung der jeweils letzten beiden Drehungen gespeichert und ausgewertet und damit verhindert, dass nach einer Drehung nach links und einer nach rechts wieder links herum gedreht wird und umgekehrt.

Signalisierung

Als passive Signalisierung für das Tracking der Roboterplattform per Kamera kommt ein planares Schachbrett-Muster zum Einsatz. Die drei Variablen, die ein solches Muster charakterisieren sind dessen Breite, Höhe und die Kantenlänge der Quadrate (Maßstab). Damit ist es ein bekanntes 3D-Objekt. Sind Höhe und Breite des Musters ungleich und so gewählt, dass einer weißen Ecke jeweils eine schwarze diagonal gegenüber liegt und umgekehrt, lässt sich ein eindeutiger Ursprung für ein lokales Koordinatensystem definieren. Maßgeblich sind die Punkte, an denen sich die Schachbrettquadrate treffen. Um diese zu finden, bietet OpenCV eine Funktion, die die Quadrate mit Hilfe morphologischer Operatoren detektiert. Da damit für die Auswertung eines Videostroms nur geringe Bildraten erreicht werden, wurde hier basierend auf einem Ansatz von Bennett und Lasenby (2010) eine eigene Funktion für die Schachbrettdetektion implementiert, die höhere Raten ermöglicht. Sie basiert auf der Filterung des Bildes mit einem Pixelring, dessen Grauwertübergänge Aufschluss darüber geben, wie wahrscheinlich das zentrale Pixel innerhalb des Kreises eine Schachbrettecke repräsentiert.

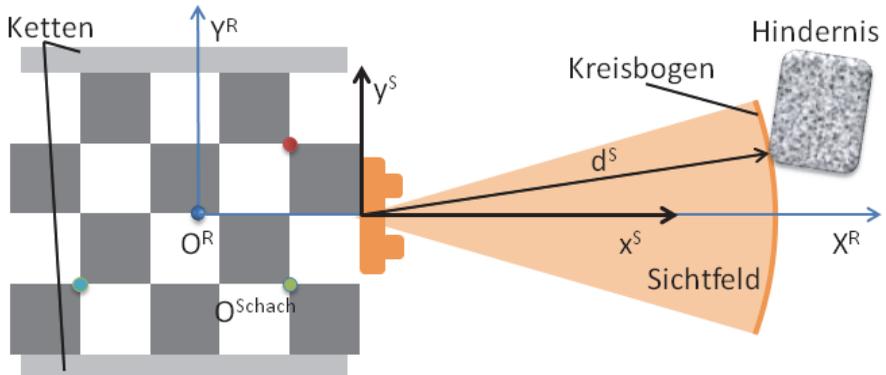


Abbildung 3: Roboter- und Sensorkoordinaten

Schematische Darstellung einer Aufsicht des Roboters mit dessen im Drehzentrum O^R gelagertem lokalem Koordinatensystem (X^R, Y^R) sowie dem im Ultraschallsensor gelagerten Sensorkoordinatensystem (x^S, y^S) . Die in letzterem gemessene Distanz d^S zu einem Hindernis im Sichtfeld des kegelförmig ausgestrahlten Ultraschalls ist die Entfernung zu dem Kreisbogen – die Richtung zu dem Hindernis ist innerhalb des Sichtbereichs unbekannt. Das Schachbrettkoordinatensystem mit dem Ursprung O^{Schach} ist angedeutet.

Koordinatensysteme

Das Schachbrett stellt ein lokales *Schachbrettkoordinatensystem* dar, das per Kamera beobachtet wird, um Position und Orientierung des Roboters in jedem Bild zu bestimmen. Parallel lassen sich Position und Orientierung des Roboters auch ausgehend von einem frei gewählten Startpunkt anhand der Radumdrehungen, die über die Zeit aufsummiert werden, ermitteln (Odometrie, Radumfang- und abstand wurden hierfür kalibriert) (Siciliano & Khatib, 2008). Das Koordinatensystem in dem dies geschieht sei das *Objektkoordinatensystem* dessen Ursprung durch die Startposition definiert ist. Um diese Positionierung mit der kamerabasierten kombinieren zu können, wird ein globales *Kamerakoordinatensystem* definiert, dessen Ursprung im Projektionszentrum der Kamera liegt (Abbildung 4). Weitere *Sensorkoordinatensysteme* werden benötigt. Ihre Lage muss innerhalb eines *lokalen Roboterkoordinatensystems* bekannt sein, um Messungen in dieses und von dort aus in Objektkoordinaten transformieren zu können. Damit wird die gesuchte Karte erstellt. Diese Systeme zeigt Abbildung 3. Das lokale Robotersystem ist im *Drehzentrum* des Roboters gelagert, dessen Position für die Transformation relativ zum lokalen Schachbrettsystem benötigt wird. Lokales Roboterkoordinatensystem und das des Schachbretts sind achsparallel und ihre X-Y-Ebene ist parallel zum Fußboden.

Ermittlung der Position und Orientierung des Roboters

Das Ziel ist nun, Position und Orientierung des Roboters in Bezug zum Objektkoordinatensystem zu jedem Zeitpunkt zu bestimmen. Um die per Kamera ermittelten Positionen und Orientierungen in dieses System zu transformieren, wird, parallel zur Definition des Objektkoordinatensystems über die Startposition des Roboters, aus 2D-2D-Punktkorrespondenzen direkt eine Homographie (Hartley & Zisserman, 2003) zwischen den Bildkoordinaten in der Bildebene und den Schachbrettkoordinaten, die alle in der Ebene des Schachbrettes liegen, berechnet. Diese bildet die Schachbrettebene auf die Bildebene ab und umgekehrt [1].

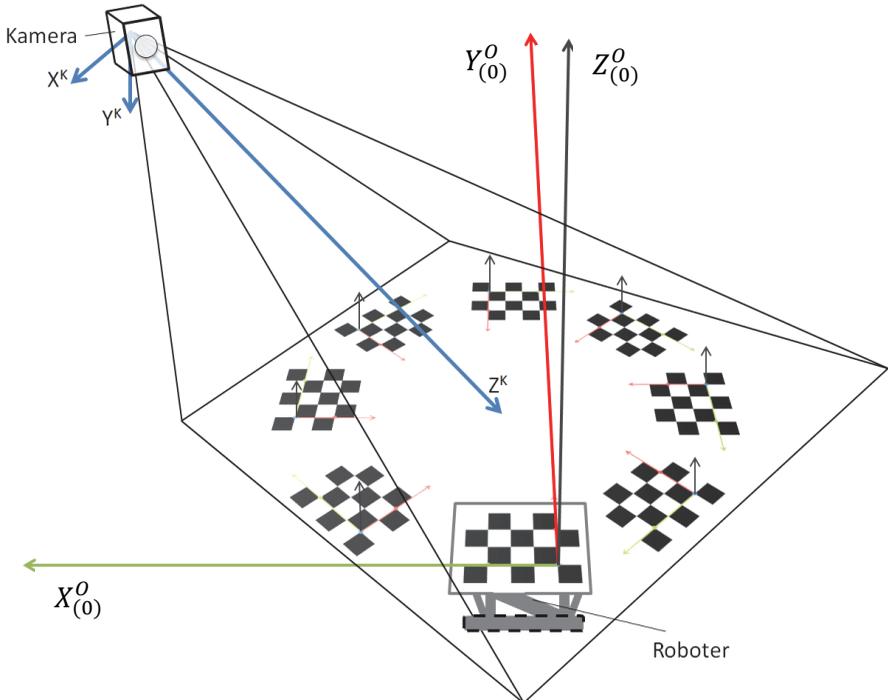


Abbildung 4: Das Kamera- und das Objektkoordinatensystem

Gezeigt sind das in der Kamera gelagerte Kamerakoordinatensystem (X^K, Y^K, Z^K) sowie einige Messungen des Schachbretts bei einer Kreisfahrt des Roboters innerhalb des Kamerasichtbereichs. Das Objektkoordinatensystem bildet das um die Z-Achse ergänzte Schachbrettkoordinatensystem $(X_{(0)}^O, Y_{(0)}^O, Z_{(0)}^O)$ zu Epoche 0, die hier mit dem Roboter skizziert ist. Die lokalen Schachbrettkoordinatensysteme für jede weitere Epoche sind ebenfalls angedeutet.

$$x_{(t_0)}^{Schach} = H_{Bild(t_0)}^{Schach} x_{(t_0)}^{Bild} \quad [1]$$

Damit ist die Lage des Schachbretts im Kamerakoordinatensystem zum Startzeitpunkt bekannt. Um nun das Tracking über die Zeit in Videobildern zu realisieren, wird diese Homographie als Transformationsvorschrift genutzt, um das in jeder Epoche auf der Bildebene abgebildete Schachbrett in die Ebene des Objektkoordinatensystems zu transformieren [2].

$$x_{(t_i)}^{Objekt} = H_{Bild(t_i)}^{Schach} x_{(t_i)}^{Bild} \quad [2]$$

Während das erste Bild zum Zeitpunkt t_0 als Nullepoche also die Startposition der Odometrie und die Homographie gemeinsam festlegt, werden alle weiteren per Kamera gemessenen Roboterpositionen über das kamerafeste System in dieses gemeinsame Objektkoordinatensystem übertragen. Dies zeigt Abbildung 4. Es gilt die Annahme, dass der Roboter sich auf ebenem Boden bewegt und das Schachbrett parallel zum Boden angebracht ist, sich also immer in einer Ebene bewegt. Position und Orientierung aus der Odometrie liegen direkt auf das Objektkoordinatensystem bezogen vor.

Kalibrierung des Rotationszentrums

Als Bezugspunkt für die Verbindung der Sensoren des Roboters mit dem optisch gemessenen Schachbrett wird, wie zuvor angesprochen, das Rotationszentrum des Roboters herangezogen. Alle Messungen werden im Objektkoordinatensystem registriert. Die Ermittlung des Rotationszentrums entspricht der Frage, wo dieses in den lokalen Schachbrettkoordinaten liegt. Es wird im Sinn einer Kalibrierung einmalig ermittelt, indem die Kamera den Ursprung des Schachbretts über eine 360°-Drehung des Roboters beobachtet. Dieser beschreibt dabei einen Kreis

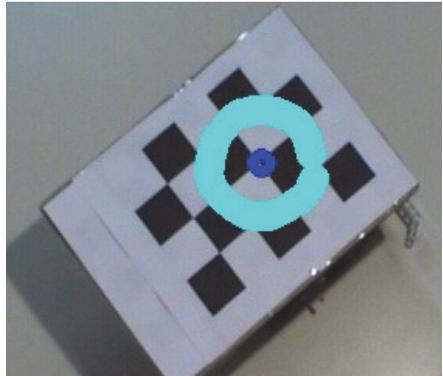


Abbildung 5:
Kalibrierung des Rotationszentrums

im Objektkoordinatensystem, dessen per Ausgleichung ermittelter Mittelpunkt dem Rotationszentrum entspricht (Abbildung 5). Damit wird die Kombination von Odometrie und Kamerapositionierung ermöglicht. Das Rotationszentrum wird herangezogen, um beide achsparallelen Systeme im Objektkoordinatensystem zusammenzuführen.

Integration der Beobachtungen

Die Kombination der Positionen aus Kameramessung und Odometrie macht ein Synchronisieren der Zeit zwischen beiden Systemen nötig. Jeder Bildmessung sowie jeder Messung eines Sensors des Roboters wird ein Zeitstempel hinzugefügt. Kamera, PC, sowie NXT besitzen jeweils ihre eigene Uhr. Der nötige kontinuierliche Abgleich zwischen Kamera und PC erfolgt über einen Network Time Protocol (NTP)-Server innerhalb des Netzwerks. Der Offset der NTP-Zeit zur Zeit des NXT-Bausteins muss nach jedem Einschalten des Systems neu ermittelt werden, da die Uhr des NXT dann von Null beginnend Millisekunden zählt. Hierfür werden Zeitstempelpaare per Bluetooth ausgetauscht.

Sowohl die von der Kamera gelieferte als auch die per Odometrie ermittelte Roboterposition und -orientierung werden auf dem PC weiter verarbeitet. Die Kombination der beiden erfolgt also dort. Dabei werden die unterschiedlichen Ausprägungen der auftretenden Fehler genutzt, um sie zu reduzieren: Die Odometrie liefert hochfrequenter Daten als die Kamera. Die Fehler zwischen den Epochen sind gering, summieren sich über die Zeit aber auf (Drift), da es sich um relative Messungen handelt. Die Kamera liefert hingegen in niedrigerer Frequenz absolute Messungen, deren Genauigkeit mit der Zeit nicht schlechter wird. Diese dienen der Odometrie als Stützstellen, indem sie den aufflaufenden Fehler zurücksetzen. Da dies in der aktuellen Implementierung auf dem PC stattfindet, kann auch erst dort, wie im nächsten Abschnitt erläutert, die Kartierung stattfinden, die für die Integration der Messwerte zunächst die Roboterposition und -orientierung benötigt. Mögliche Aussetzer beider Methoden werden abgefangen. Verlässt der Roboter den Sichtbereich der Kamera, liefert die Odometrie weiterhin Werte, wobei die Drift sich dann nicht mehr korrigieren lässt.

Kartierung

Die durch den Roboter erfassbaren Informationen stammen von dessen Stoßstange und seinem Ultraschallsensor. Hinzu kommt die Information, dass die Position, an der sich der Roboter befindet, frei sein muss. Die auf die jeweils aktuelle korrigierte Position bezogenen Messungen werden in einer Rasterkarte zusammengetragen, die Auskunft darüber gibt, ob eine Rasterzelle belegt, frei oder unbestimmt ist. Für die Markierung der aktuellen Roboterposition als unbelegt, ist das direkt möglich. Für den Tastsensor muss die Lage der Stoßstange relativ zur Drehachse des Roboters in dem lokalen Roboterkoordinatensystem vorliegen. Sie ist mittig in einem festen Abstand vom Drehzentrum in Fahrtrichtung angebracht. Auch der Ultraschallsensor ist so montiert, dass er in Richtung der x-Achse des Roboters in Fahrtrichtung blickt. Abbildung 3 zeigt dies in der Aufsicht inklusive einem in dem Sensor gelagerten Sensorkoordinatensystem und einem Hindernis, dessen Distanz aufgrund der Ultraschallausbreitung als Kreisbogen gemessen wird.

Sensormessungen tragen dazu bei, die Häufigkeit von Rasterzellen zu erhöhen oder zu verringern. Häufigkeiten kleiner Null werden als frei, solche größer Null entsprechend als belegt interpretiert. Die Zellen, in denen sich der Roboter aufhält, werden durch verringern der Häufigkeit als unbelegt markiert. Der Tastsensor tut dies für belegte Zellen und die Ultraschallmessung steuert beides bei: Zellen auf dem Kreisbogen sind potentiell belegt, die, die das Signal auf dem Weg dorthin überstreicht, gelten als frei. Es wird außerdem angenommen, dass Zellen, die der Roboter durchquert, mit höherer Wahrscheinlichkeit frei sind als solche, in denen per Ultraschall nichts detektiert wurde. Die Markierung einer Zelle als belegt ist per Ultraschall unsicherer als die mittels Tastsensor.

Das Gebiet muss mit dieser Vorgehensweise mehrfach mit Messungen abgedeckt werden, um ausreichend hohe Häufigkeiten zu erhalten, die für oder gegen eine Belegung der jeweiligen Zellen sprechen.

4 Ergebnisse

Untersuchungen der eingesetzten Signalisierung mit einem 4x3-Schachbrettmuster ergeben eine mittlere Standardabweichung von 0,3 mm für die wiederholte Detektion eines ruhenden Schachbretts in der Testumgebung. Eine Quaderkante verläuft dabei in der Länge durch etwa 30 Pixel.

Mit dem Kettenantrieb variiert das ermittelte Rotationszentrum des Roboters sowohl auf Teppich als auch auf glattem Holz im Bereich eines halben Zentimeters und ist damit deutlich ungenauer als die Bildmessungen des Schachbretts. Ursache sind verschiedene Fehler, allen voran ein unkontrolliertes Rutschen der Ketten, das bei der Drehbewegung besonders ausgeprägt ist. Aus diesem Grund wurden testweise die Ketten des Roboters durch Gummiräder ersetzt, was zu einer Standardabweichung des mehrfach gemessenen Zentrums im Millimeterbereich führt.

Eine Trajektorie über 8 Minuten und die dabei erhaltene Rasterkarte zeigen die Abbildungen 6 und 7. Es zeigt sich, dass freie Zellen durch eine hohe Gewichtung der Markierung der Position des Roboters als frei, sehr große Häufigkeiten haben. Liegen, wie im Randbereich, nur Messungen des Ultraschalls vor, nehmen Unsicherheiten zu, die von der Oberflächenbeschaffenheit der Hindernisse abhängen, auf die das Signal trifft. Unten links findet sich beispielsweise glatter Kunststoff, der das Signal je nach Auftreffwinkel spiegelt, so dass Mehrfachreflexionen auftreten

können; unten rechts hingegen Pappkartons, die das System sicherer in der richtigen Distanz detektiert. Auch nach 8 Minuten ist das Gebiet ungleichmäßig erfasst.



Abbildung 6: Trajektorie aus Kamerasicht über 8 Minuten

Jeder Kringel repräsentiert den Ursprung des Schachbretts in einem Bild. Diese wurden über die Zeit in das aktuelle Kamerabild geplottet.

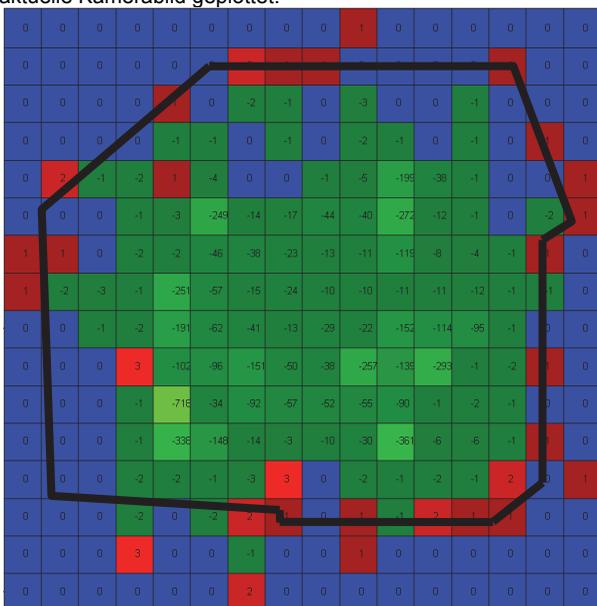


Abbildung 7: Karte nach 8 Minuten

Die Abbildung zeigt die aus der in Abbildung 1. gezeigten Fahrt entstandene Karte. Hellere Farbtöne stehen für eine höhere Wahrscheinlichkeit, die auch anhand der eingetragenen Werte pro Rasterzelle ermittelt wurde. Grün = frei (<0), blau = unbestimmt/unsicher (=0), rot = belegt (>0). Die Sollgrenzen sind in schwarz dargestellt.

5 Ausblick

Im Sinne eines Ausblicks finden sich viele Ansatzpunkte. Bereits teilweise umgesetzt ist die Erweiterung des Systems um die Ausnutzung der PTZ-Funktionalität der Kamera. Abbildung 8 zeigt die Ergänzung der Fahrstrategie um die Kamera als Sensor, der den Roboter stoppen, sich neu ausrichten und diesen nach Fortführung der initialen Homographie weiterfahren lässt. Hierbei ist die Änderung der inneren und äußeren Orientierung zu berücksichtigen. Eine automatische simultan ablaufende Kamerakalibrierung könnte hier Abhilfe schaffen. Weiterhin kann die Nutzung mehrerer Kameras ein nächster Schritt sein.

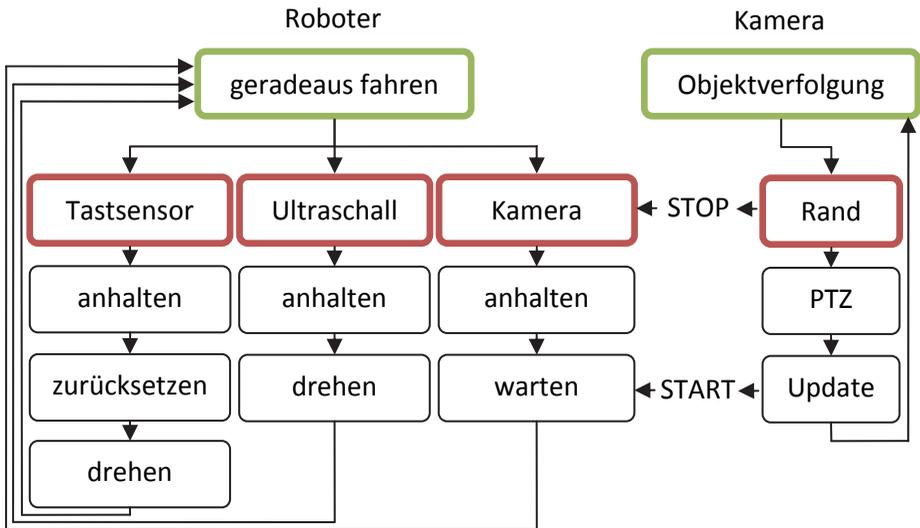


Abbildung 8: Fahrstrategie mit PTZ-Nutzung

Ergänzung der Kamera als dritten Sensor, der das Fahrverhalten beeinflussen kann. Kommt der Roboter dem Rand des Sichtbereichs zu nah, wird er gestoppt. Es folgt eine Neuausrichtung der Kamera ggf. inklusive Zoom, während der der Roboter wartet. Liegt die neue initiale Homographie vor, setzt er seine Fahrt fort.

Die Erkennung des Schachbretts basiert bisher auf den Einzelbildern des Videos. Die Bildsequenz wird noch nicht ausgenutzt, um beispielsweise die Detektion zu beschleunigen. Eine iterative Ausgleichung, die 3D-2D-Punktkorrespondenzen nutzt, könnte bei der Transformation ins kamerafeste System die Homographie ersetzen und so die Fehlerquelle minimieren, die die Annahme, Boden- und Schachbrettebene seien parallel, mit sich bringt. Denn jeder Schachbrettstück liegt in 3D-Schachbrettkoordinaten und in 2D-Bildkoordinaten vor. Das Ergebnis eines damit durchgeführten Rückwärtsschnitts sind die 6 Parameter der äußeren Orientierung der Kamera bezogen auf das Schachbrettkoordinatensystem. Außerdem sind alternative Signalisierungsmöglichkeiten zu prüfen.

Der Maßstab der Bildmessung kommt aus der Kantenlänge des Schachbretts. Da dieses nur einen kleinen Teil des Bildes abdeckt, könnte eine gefahrene Strecke einen längeren und damit zur Szene passenden Maßstab liefern. Die Streckenmessung per Odometrie mit Rädern ist bei einer Geradeausfahrt auf etwa 0,5% genau.

6 Zusammenfassung

Es wird eine selbst entwickelte Roboterplattform vorgestellt, die mittels Tast- und Ultraschallsensoren autonom ihre Umgebung erkundet und sich per Odometrie selbst lokalisiert. Dies geschieht in einer Testumgebung, die von einer Überwachungskamera aufgenommen wird. Diese sieht ein auf der Roboterplattform montiertes Schachbrettmuster und ermöglicht so die Ermittlung deren 3D-Position. Beide Positionsquellen werden kombiniert. Der gesamte Ablauf umfasst das Herstellen der äußeren Orientierung zwischen der Kamera und dem im Roboter gelagerten Objektkoordinatensystem und läuft ohne manuelles Eingreifen ab. Als Ergebnis wird eine Rasterkarte geliefert, die es ermöglicht, häufigkeitsbasiert freie Zellen von belegten zu unterscheiden. Außerdem wird ein Verfahren vorgestellt, das initial automatisch das Drehzentrum des Roboters mithilfe der Beobachtungen der Überwachungskamera per Ausgleichung ermittelt.

Mit der Karte liegt ein Produkt vor, das verschiedensten Szenarien der Videoüberwachung und –auswertung als Eingangsdatensatz dienen kann. Beispielsweise liefert sie Anhaltspunkte für mögliche Objektverdeckungen beim Tracking und Tiefen- sowie geometrische Informationen über die Szene.

Neben der Ausweitung des Sichtbereichs durch die Nutzung der PTZ-Funktion der Kamera, bieten sich sowohl von Seiten der Hardware als auch von der Software Ansatzpunkte für zukünftige Arbeiten, um zeitliche sowie geometrische Auflösung zu erhöhen. Insbesondere ist eine Erweiterung auf eine 3D-Kartierung beispielsweise durch einen 3D-Laserscanner oder ein auf dem Roboter montiertes Stereokamerasystem vorstellbar. Des Weiteren ist die aktuelle Fahrstrategie um die Nutzung der selbst erfassten Karte schrittweise erweiterbar, um die Wahl der Trajektorie nicht dem Zufall überlassen zu müssen.

7 Literaturverzeichnis

Bennett, S., & Lasenby, J. (2010). *ChESS – Quick and Robust Detection of Chessboard Features*. Cambridge: Cambridge University.

Berns, K., & Schmidt, D. (2010). *Programmierung mit LEGO MINDSTORMS NXT*. Heidelberg: Springer.

Hartley, R., & Zisserman, A. (2003). *Multiple View Geometry in Computer Vision* (2. Ausg.). Cambridge: Cambridge University Press.

Luhmann, T. (2010). *Nahbereichsphotogrammetrie: Grundlagen, Methoden und Anwendungen* (3. Ausg.). Berlin: Wichmann.

Siciliano, B., & Khatib, O. (2008). *Springer Handbook of Robotics*. Berlin: Springer.