

Bau eines Spinnenroboters

Bericht zum Projekt des Lego Mindstorms Seminars

Philipp Schümann, ET-IT
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Moderne Roboter verwenden immer öfter Bilderkennung um sich zu Orientieren und ihre Aufgabe zu erfüllen. Durch immer höhere Kameraauflösungen können selbst kleine Details besser erkannt werden. Die Auswertung von Bild-daten verbraucht viel Rechenleistung und nicht immer müssen kleinste Details erkannt werden. Im Folgenden wird versucht eine Möglichste schnelle Zielerfassung mithilfe von Bildverarbeitung zu realisieren. Die Bilder werden durch ein Gütekriterium in der Größe variiert, um so die Auswertung zu beschleunigen.

Schlagwörter—Bildauswertung, Lego, Matlab, Spinne, Zielerfassung

I. EINLEITUNG

DIE Bildauswertung wird bei Robotern immer wichtiger. Vor allem in der Industrie ermöglicht diese dem Roboter immer komplexere Aufgaben zu lösen, doch auch privat kommen immer häufiger Roboter mit Bildauswertungstechnik zum Einsatz. In der Industrie ermöglichen Roboter das Arbeiten an gefährlichen Orten, ohne dabei Menschen zu gefährden. Auch ist Bildauswertung an Robotern sehr gut geeignet um das Zusammenarbeiten zwischen dem Roboter und Menschen zu ermöglichen.

In dem Projekt im Rahmen des LEGO Mindstorms Seminars wurde ein Roboter entworfen und programmiert, der in der Lage sein sollte, auf ein Ziel, welches mithilfe von Bildauswertung erkannt werden sollte, sich auszurichten und zu schießen.

Die Konstruktion des Roboters in Form einer Spinne kann für unwegsames Gelände verwendet werden, in dem normale Roboter mit Ketten oder Rädern sich nicht fortbewegen können.

Der Roboter wurde aus LEGO gebaut, zum Antrieb des Roboters wurden ein NXT-Baustein und drei NXT-Motoren verwendet. Programmiert wurde der Roboter in Matlab mit der Toolbox der RWTH Aachen [3]. Zusätzlich zu Legoteilen wurde eine IP-Webcam in Form eines Handys sowie eine kleine Batteriebox verbaut.

II. VORBETRACHTUNGEN

Die Idee zu dem Spinnenroboter hat sich aus zwei Teilideen zusammengesetzt. Diese werden hier kurz vorgestellt.

A. Roboter mit Zielerkennung

Bereits im vorherigen Jahr haben Studenten im Rahmen des Lego-Projektes einen Roboter gebaut, der sich selbstständig auf ein Ziel ausrichtet. Dieser ist in Abbildung 1 zu sehen. Das wurde damals mit Ultraschallsensoren zur Zielerfassung umgesetzt. Ultraschallsensoren haben nur ca. 25 cm Reichweite. Dies war uns zu wenig, deshalb wurde anstatt der Ultraschallsensoren eine IP-Webcam verwendet.

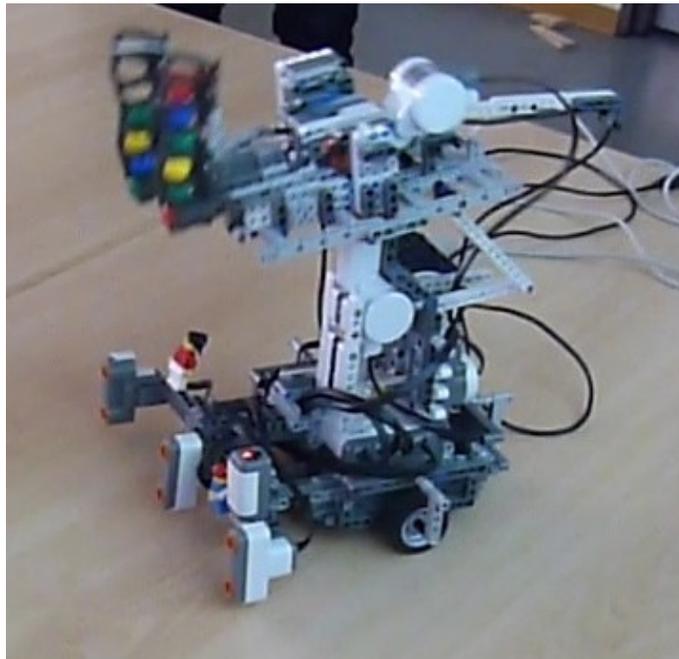


Abbildung 1. Legoroboter aus dem Mindstorms-Seminar 2017 [2]

B. Spinnenroboter

Das Design des Roboters wurde durch den Film "Wild Wild West" inspiriert. In dem Film wurde von Wissenschaftlern eine sehr große schwer bewaffnete Metallspinne gebaut. In Abbildung 2 ist ein Nachbau der Spinne mit Lego zu sehen. Diese ist aber nicht mechanisiert.

III. HAUPTTEIL

A. Konstruktion

Während der Umsetzung des Projektes wurde schnell klar, dass beim Design ein paar Abstriche gemacht werden mussten, so hat das Spider-Vehicle nur 6 anstatt der für eine Spinne üblichen 8 Beine bekommen. Diese Beine konnten auch keine Gelenke bekommen, sondern sind nur starr gerade. Gelenke hätten die Stabilität der Beine verringert und auch fehlten die Steuermöglichkeiten für diese. Ein Gesamtbild des Roboters ist in Abbildung 3 zu sehen. Bei dem Roboter war vor allem die Stabilität entscheidend, da das ganze Gewicht nur auch den 6 Beinen lag, von denen meistens nur 3 Beine auf dem Boden standen. Die anderen 3 Beine berührten den Boden meist aufgrund der Laufbewegung nicht. Die Unterseite der Füße ist mit Gummi ausgerüstet wie in Abbildung 4 zu sehen ist. Diese verhindern das Rutschen der Füße bei



Abbildung 2. Lego Spinnenroboter nachgebaut aus dem Film "Wild Wild West"[1]

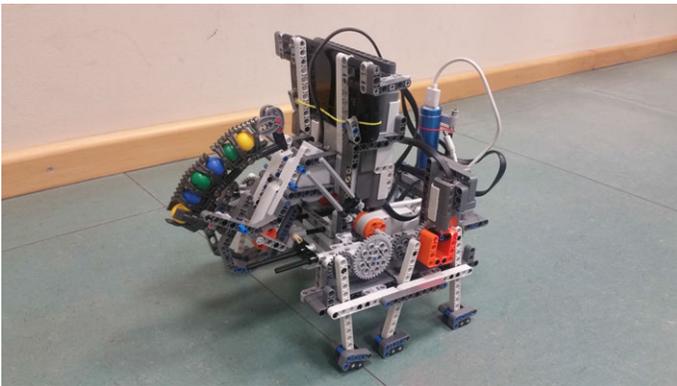


Abbildung 3. Roboter



Abbildung 4. Roboterfüße

Lauf- oder Drehbewegungen des Roboters. Auch die Präzision der Bewegungen konnte damit erhöht werden. Der nächste Punkt, der mit der Stabilität des Roboters einhergeht ist der Schwerpunkt des Roboters. Aufgrund der vorne angebauten Kanone, sowie der IP-Webcam in Form eines Handys, musste ein entsprechendes Gegengewicht geschaffen werden. Das wurde durch die Befestigung der Motoren, sowie eine kleine Batteriebox hinten am Roboter ausgeglichen. Die Batteriebox hat zudem noch den Vorteil, dass die IP-Webcam noch mit

zusätzlichem Strom versorgt wird. Die Kanone ist so angebracht, dass sie in einem 45° Winkel schießt, siehe Abbildung 5. Der 45° Winkel ermöglicht Schüsse mit 150 cm Reichweite. Der waagerechte Schuss dagegen hatte nur eine Reichweite von 40 cm. Da ein NXT auf drei Motoren beschränkt ist, war es nicht möglich die Kanone im Schusswinkel Variabel zu machen. Das Ziel des Roboters wurde im Design eines Panzers

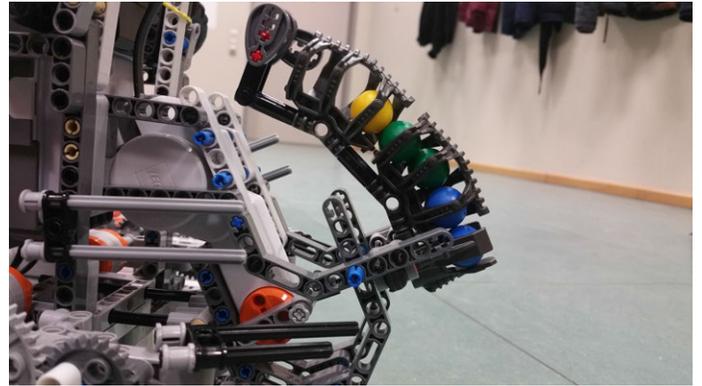


Abbildung 5. Kanone des Spinnenroboters

gestaltet wie in Abbildung 6 zu sehen ist. Diese Form wurde einerseits aus Designgründen gewählt, aber auch um die Größe des Ziels zu vergrößern und damit die Wahrscheinlichkeit das Ziel nicht zu treffen zu verringern.

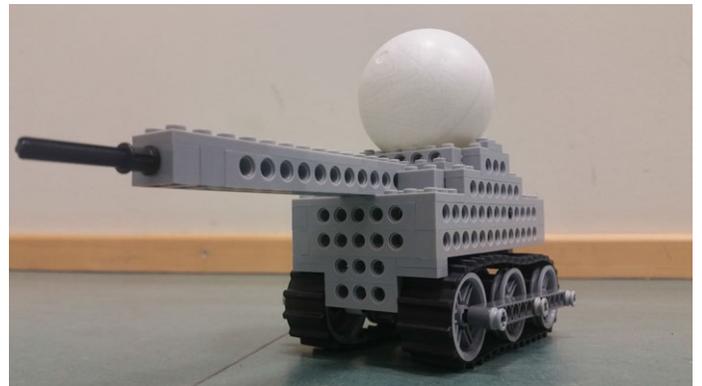


Abbildung 6. Roboterfüße

B. Programmierung

In Bild 7 ist die grafische Benutzer Schnittstelle zu sehen. Links oben befinden sich die Steuerelemente nur manuellen Steuerung des Roboters. Wenn eine Bewegungsaktion aktiviert wird, bewegt sich der Roboter so lange, bis entweder eine andere Bewegungsaktion gestartet wird, oder der Roboter mithilfe des Stop-Buttons angehalten wird. Der Stopp-Button stoppt den Roboter aber nicht direkt, da dieser sich bei einem Stopp in eine stabile Standposition begibt. Durch Drehungen des Roboters ist es möglich, dass die Füße eine Stellung einnehmen, in der der Roboter umkippen kann. Auf der rechten Seite sieht man das Bild von der IP-Webcam. Auf dem Bild ist mit dem blauen Kreis das aktive Ziel markiert, alle weiteren

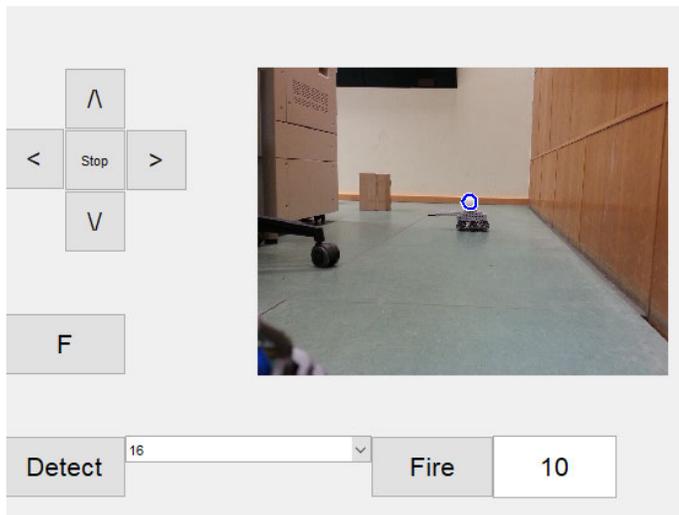


Abbildung 7. GUI

möglichen Ziele würden mit einem roten Kreis markiert werden.

Die unteren Schaltflächen sind für die automatische Zielfindung. Mit dem Detect-Button wird ein aktuelles Bild von der IP-Webcam abgerufen, alle Ziele werden auf dem Bild markiert und dann rechts in der GUI angezeigt. In dem Dropdown-Menü werden alle möglichen gefundenen Ziele angezeigt. Das ausgewählte Ziel wird blau markiert. Die Namen für die Ziele sind Angaben zur Größe des Ziels, sodass die Zielauswahl dadurch erleichtert wird.

Im Input-Feld ganz rechts kann die Anzahl der Schüsse, die auf das Ziel abgegeben werden, eingegeben werden. Der letzte Button "Fire" startet den Algorithmus, mit dem sich der Roboter auf sein Ziel ausrichtet.

Der Algorithmus startet immer mit einem neuen Abruf von der IP-Webcam. Auf dem Bild wird das ausgewählte Ziel erneut gesucht, auf Basis von der Position und Größe des Ziels, das zuvor ausgewählt wurde.

Die Bildverarbeitung übernimmt die Imageprocessing-Toolbox in Matlab. Die Funktion *imfindcircles* markiert alle kreisförmigen Objekte auf dem Bild und gibt deren Mittelpunkt und Radius in einer Matrix zurück. Da die Zielfindung über die Form und nicht über die Farbe läuft muss die Funktion nur zwei mal aufgerufen werden, um alle kreisförmigen Ziele zu finden.

Die Funktion wird einmal für Objekte mit hellerer Farbe als der Hintergrund und einmal mit dunklerer Farbe als der Hintergrund aufgerufen. Das Bild wird vor der Auswertung in ein Graustufenbild umgewandelt. Um nicht das Ziel bei fälschlicherweise neu gefundenen Zielen zu verlieren wird nicht das vollständige Bild ausgewertet, sondern nur ein Teil des Bildes auf Basis eines Gütekriteriums der vorherigen Bewegung. Sollte kein Ziel mit dem aktuellen Gütekriterium gefunden werden wird das Gütekriterium verschlechtert. Nach fünf Fehlschlägen in Folge wird der komplette Zielfindungsalgorithmus abgebrochen und muss neu gestartet werden.

Die Sollposition des Ziels ist die Mitte des Bildes und ist somit bekannt. Für die ersten Schritte des Algorithmus wird

auch nur die x-Achse des Bildes beachtet und somit wird die Höhe des Ziels ignoriert. Sollte das Ziel sich links von der Sollposition befinden dreht sich der Roboter nach links, analog dazu, wenn sich das Ziel rechts von der Sollposition befindet, dreht sich der Roboter nach rechts. Sobald sich das Ziel im mittleren Drittel des Bildes befindet, geht der Algorithmus zu Schritt zwei über und das Gütekriterium wird verbessert, sollte das nicht der Fall sein, wird der aktuelle Schritt wiederholt.

Im zweiten Schritt des Algorithmus wird der Abstand zum Ziel angepasst. Der Abstand wird aus der Bildgröße im Vergleich zur bekannten Größe des Ziels bestimmt. Ist das Ziel zu klein auf dem Bild, bewegt sich der Roboter auf das Ziel zu. Wieder analog dazu bewegt sich der Roboter rückwärts, wenn das Ziel zu groß auf dem Bild ist. Falls Aufgrund der Vorwärts- oder Rückwärtsbewegung das Ziel das mittlere Drittel des Bildes verlässt, wird das Gütekriterium verschlechtert und der erste Schritt wird erneut ausgeführt. Wenn die Entfernung des Ziels innerhalb des zulässigen Bereichs liegt, ist die grobe Ausrichtung abgeschlossen, das Gütekriterium wird erneut verbessert und die Feinausrichtung beginnt.

Die Feinausrichtung ist relativ langsam. Aufgrund der Konstruktion des Roboters mit den Beinen sind minimale Bewegungen schwierig zu realisieren, da sich bei einer Drehung auch immer der Abstand der Kanone zum Boden verändert, sowie auch der Winkel der Kanone variiert je nach Position der Beine. Der Abstand der Kanone zum Boden variiert nur wenig und kann daher vernachlässigt werden. Die Variation des Winkels der Kanone muss aber beachtet werden, da die Kanone sonst sehr leicht über das Ziel schießen kann, oder auch nur den Boden vor dem Ziel treffen kann. Da der Winkel der Kanone nicht wirklich verändert werden kann, wurde ein Bereich definiert, in dem ein Treffen des Ziels möglich ist. Der Winkel wird aus dem Rotationssensor der IP-Webcam ausgelesen. Die IP-Webcam bietet eine REST-API Schnittstelle, um deren Sensordaten im JSON-Format auszulesen. Vor jedem Abruf von der IP-Webcam, Sensordaten und Bilder, muss bei der Feinausrichtung eine kurze Pause eingelegt werden, da sonst aufgrund der Verzögerung der IP-Webcam bzw. der Übertragung ein nicht aktuelles Bild oder ein verschwommenes Bild übertragen wird.

Die Feinausrichtung ist nochmals in 3 Unterstufen unterteilt, abhängig davon wie nah das Ziel an der Sollposition ist. Auf 150 cm Reichweite mit einer Kugel mit 1 cm Radius ein Ziel zu treffen, das einen Radius von 3 cm hat. Ein Grad Abweichung ergibt auf 150 cm eine seitliche Verschiebung von 3 cm, auf Grund von der Größe des Ziels steht damit auch der Zielbereich fest, da die maximale Abweichung, mit der das Ziel noch getroffen werden kann, somit bei $\pm 1\%$ liegt. Da dieser Bereich sehr klein ist und aufgrund von der Konstruktion des Roboters mit 6 Beinen dreht sich der Roboter mehrfach über das Ziel, verbessert dabei aber immer die Distanz zwischen der Soll- und Istposition des Ziels. Wenn eine Ausrichtung gefunden wurde wird der Winkel der Kanone mithilfe der Sensordaten von der IP-Webcam geprüft. Sollte der Winkel außerhalb des gültigen Bereichs liegen wird versucht diesen zu verbessern, indem sich der Roboter rückwärts oder vorwärts bewegt, da dadurch sich die Zielposition möglichst wenig verändert. Wenn das Ziel trotzdem nicht mehr im Bereich $\pm 1\%$ liegt, wird die

Feinausrichtung fortgesetzt. Jeder Zyklus der Feinausrichtung dauert inklusive Bewegung des Roboters ca. 4 Sekunden.

Da der Datenverkehr zwischen der IP-Webcam und dem Rechner sehr hoch ist, wurde ein eigenes kleines Netzwerk für diesen Zweck eingerichtet. Durch das private Netzwerk ist es auch sichergestellt, dass niemand sonst zugriff auf die IP-Webcam des Roboters bekommt.

IV. ERGEBNISDISKUSSION

Das Endergebnis ist ein Spinnenroboter mit 6 Beinen. Die Gewichtsverteilung und Stabilität des Roboters konnten ausreichend optimiert werden, obwohl die baulichen Möglichkeiten begrenzt waren.

Aufgrund der baulichen Begrenzungen ist unser Spinnenroboter nicht Geländetauglich. Die Schritthöhe ist dafür zu niedrig und auch haben die Beine keine Gelenke.

Mithilfe der GUI kann der Roboter manuell gesteuert werden, da der Roboter über Bluetooth verbunden ist und über eine IP-Webcam verfügt muss man den Roboter selbst beim Steuern nicht unbedingt im Blick haben.

Die GUI kann auch die automatische Zielerfassung starten. Diese hat beim Schießen mit hoher Zuverlässigkeit das Ziel getroffen.

Der Roboter war in der Lage die ihm gestellte Aufgabe, das Suchen und Abschießen eines Ziels, zu erfüllen.

V. ZUSAMMENFASSUNG UND FAZIT

Im Rahmen des Legoprojektes ist es gelungen einen Spinnenroboter zu bauen, der in der Lage ist, ein Ziel mithilfe eines Bildes zu erfassen und dieses abzuschießen. Die Zielerfassung läuft über ein mehrschrittiges System mit einem Gütekriterium um den Rechenaufwand und somit auch die Rechenzeit zur Bildauswertung zu reduzieren. Die Bildauswertung könnte unter Zuhilfenahme eines Grafikprozessors noch weiter beschleunigt werden. Die Zielerfassung funktioniert nur für stationäre Ziele die durch einen Kreis oder eine Kugel markiert sind. Die Zielerfassung ist somit noch verbesserungswürdig, dies Würde aber über den Rahmen des Praktikums hinausgehen und auch die Motoren wären für diesen Schritt nicht gut geeignet. Allgemein war die Konstruktion des Roboters in Form einer Spinne nur wenig geeignet für eine Feinausrichtung, mit einem Kettenfahrzeug wäre eine Ausrichtung einfacher möglich gewesen, aber es wurde mit dem Spinnenfahrzeug realisiert.

VI. QUELLEN

- [1] Lego Mechanische Spinne von Dr. Loveless
https://i2.wp.com/farm8.staticflickr.com/7354/11898981874_beca3b88bd_z.jpg
Version: 2018.03.23
- [2] Lego Roboter aus dem Mindstorm-Seminar 2017
<https://i.ytimg.com/vi/E7v3r9kEaco/maxresdefault.jpg>
Version: 2018.03.23
- [3] RWTH Aachen Mindstorms NXT Toolbox
<http://www.mindstorms.rwth-aachen.de/>
Version: 2018.03.23