

Bau eines flächenberechnenden Scanners

Chris-Marvin Hamann, Elektro- und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Abstract— In dem LEGO Mindstorms Projekt welches von der Otto-von-Guericke-Universität durchgeführt wurde, sollte man sich selbst kreativ betätigen, in dem sich jede Gruppe eine individuelle, praxisbezogene Anwendung der gegebenen Bauteile ausdenkt und diese dann auch umsetzt. Bezogen auf diese Aufgabenstellung haben wir uns in unserer Gruppe entschlossen einen Scanner zu entwerfen, jedoch wurden schon in den Vorjahren des Projekts Scanner gebaut, welche zum Beispiel Bilder eingescannt haben, somit wurde innerhalb unserer Gruppe beschlossen automatische Flächenberechnung und –erkennung als neue Funktionen hinzuzufügen. Im folgenden Beitrag wird beschrieben wie diese Idee konstruktionstechnisch, sowie auch in Hinsicht auf das Programmieren umgesetzt haben. Nachfolgend wird diskutiert, inwiefern diese Idee besser umsetzbar gewesen wäre, so dass mögliche Fehler minimiert werden könnten. Dazu wird genauer auf den Aufbau des Scanners eingegangen und die Flächenerkennung anhand unserer Programmierung und verschiedenen Messwerten näher erklärt und wodurch dabei Fehler aufgetreten sind.

Schlagwörter—Flächenberechnung, Formenerkennung, Lego, Mindstorms, Matlab, Scanner

I. EINLEITUNG

SCANNER gewinnen in der heutigen Gesellschaft zunehmend an Bedeutung, so findet man sie nicht nur in der Industrie, sondern auch schon im Alltag. So besitzt ein Großteil der Bevölkerung einen Drucker und in diesen ist meist auch gleich eine Scanfunktion mit enthalten, diese kann zum Beispiel auch für das Universitätsleben nützlich sein, da es vorkommen kann, das man im Krankheitsfall die Aufzeichnungen eines Kommilitonen nutzen möchte oder einfach ein Skript für einen Vorlesung braucht. Manchmal fällt es einem somit schwer Drucker beziehungsweise Scanner aus dem Alltag wegzudenken. In der Industrie ist wohl der größte Nutzen von Scannern das Ein- und Auslesen von Daten, wie zum Beispiel bei einfachen Barcodescannern. Da in unserem Fall der Scanner jedoch zum Flächenberechnung und –erkennung genutzt werden soll, könnte man sich dies zum Beispiel im technischen zeichnen vorstellen und entworfenen Räume im Maßstab zu berechnen oder auch in der Logistik um eventuell Pakete nach ihrer Größe und Form zu sortieren und somit eine bessere Lagerung zu ermöglichen.

In dem Projekt, welches ich zusammen mit Leander Bartsch bearbeitete und im Rahmen des LEGO Mindstorms Seminars vom 12. Februar bis zum 23. Februar 2018 entstand, wollten wir nun einen wie oben beschriebenen Scanner bauen. Somit

haben wir unter Betrachtung der vorgegebenen Bauteile und des beschränkten Arbeitszeitraum beschlossen, einen Scanner zu bauen, welcher ein Blatt mit von uns ausgedruckten geometrischen Formen, was in diesem Fall Kreise, Quadrate und gleichseitige Dreiecke umschloss, automatisch einzuziehen und durch einen auf der Konstruktion fahrenden motorbetriebenen Lichtsensor einzuscannen. Der angeschlossene Computer soll dann die ermittelten Messwerte auswerten und über verschiedene Schritte, welche im Hauptteil ausführlicher beschrieben werden, den Flächeninhalt der Fläche erkennen und um welche geometrische Form es handelt. Der Scanner wurde unter Verwendung des uns zur Verfügung gestellten NXT-LEGO-Bausteins gebaut und daraufhin mit Matlab programmiert. Das gescannte Bild, sowie auch die einzelnen Schritte, wie berechneter Flächeninhalt und erkannte Form, sollten auf einem Interface auf dem Bildschirm des Rechners ausgegeben werden.

II. VORBETRACHTUNGEN

Wie im vorherigen Abschnitt bereits erwähnt, gab es in früheren Jahrgängen innerhalb dieses Seminars schon Versuche Scanner zu bauen. Da uns die Informationen zu diesen Scannern mündlich von unseren Gruppenleitern weitergetragen wurden, sind sie nicht sonderlich detailliert und nicht belegt, den zweiten Scanner kann man in einem unter [1] im Literaturverzeichnis angegebenen Link sehen.

Beide Scanner hatten, im Gegensatz zu unserem, das Ziel ein Schwarz-Weiß Bild einzuscannen und danach digital wiedergeben zu können. Der Unterschied der beiden lag vor allem in der Art und Weise in der der Lichtsensor zum Scannen transportiert werden sollte.

A. Lichtsensor an hydraulischem Arm befestigt

In diesem Beispiel aus dem Jahr 2016, ist der Lichtsensor an einem Arm befestigt, welcher gleichmäßig hin und her fährt und durch einen Motor betrieben ist. Eine Schwierigkeit hierbei ist gewesen, dass sich der Arm in gleichmäßiger Geschwindigkeit bewegen soll. Dazu musste nämlich eine Funktion aufgestellt werden, welche von den damaligen Projektteilnehmern empirisch ermittelt werden musste. Dies ist notwendig, da es sonst zu gewissen Messungenauigkeiten kommen würde, was vor Allem beim Einscannen eines Bildes fatal wäre, da dort sehr genaue Messwerte benötigt werden. Man sieht außerdem, dass sie den Lichtsensor so eingestellt haben, dass er im Aktivmodus ist, wo er selbst Licht ausgibt und das reflektierte Licht misst. Für das Einscannen eines

Schwarz-Weiß Bildes ist dies optimal, da die gemessenen Lichtwerte die Schwarz- und Weißinhalte des Bildes wiedergeben können. Bei bunten Bildern müssen nämlich die Blau-, Grün- und Rotanteile des Bildes ermittelt werden, was ein deutlich größerer Rechenaufwand wäre und mit dem NXT dementsprechend lange dauern würde. Zudem ist der Einfluss äußerer Lichtquellen dadurch abgeschwächt. Diese Umsetzung soll, nach Angaben unserer Projektleiter sehr detailliert gewesen sein.

B. Lichtsensor fährt auf Zahnschienen

Dies ist die Anwendung welche auch wir genutzt haben, dort befindet sich der Lichtsensor direkt mit dem Motor verbunden auf Zahnschienen. Der Motor betreibt dann Zahnräder durch sich der Lichtsensor dann bewegt. Der Vorteil bei dieser Methode ist, dass man für eine gleichmäßige Bewegung lediglich eine bestimmte Geschwindigkeit einstellen muss. Man muss jedoch darauf achten, dass die Zahnräder auch fest auf die Schienen eingespannt sind, da es sonst zu Ungenauigkeiten kommen kann. Dieser Scanner soll nach den Aussagen unserer Gruppenleiter ziemlich ungenau gewesen sein, dies kann natürlich an Konstruktionsfehlern liegen oder daran, dass die Ungenauigkeiten, die durch die Transportationsart entstehen, zu groß sind um Bilder einzuscannen, da die Unterschiede zwischen den Grautönen relativ klein sei können und dadurch nicht akkurat erfasst

einstellen, so dass die Scangenaugigkeit dadurch nicht beeinträchtigt wurde.

Wie in den Vorbetrachtungen erwähnt, haben auch wir den Lichtsensor auf Zahnschienen fahren lassen, des Weiteren haben wir zwei Tastsensoren genutzt, damit der Motor stoppt und dann in die andere Richtung weiter fährt. Je nach dem in welche Richtung der Lichtsensor fährt, werden die Werte auch spiegelverkehrt dargestellt, weshalb wir im Programm implementiert haben, dass die jeweils zweite Reihe gedreht wird.

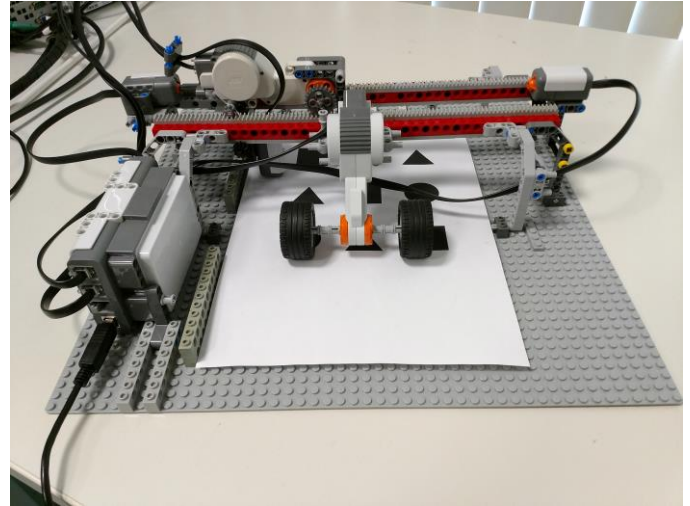


Abbildung 2: Scanner mit eingezogenem Blatt

Während der Lichtsensor hin- und herfährt, werden die reflektierten Lichtwerte gemessen, welche man in der Abbildung 1 in einem Diagramm dargestellt sehen kann. Dort kann man sehen dass die Werte von 0 bis 1023 gehen, dabei ist null sehr dunkel, also schwarz und 1023 weiß. Durch die vor Ort gegebenen Helligkeitsverhältnisse liegen die Werte welche Schwarz entsprechen bei circa 400 bis 500 und für Weiß bei 700 bis 800. Das Programm sucht dann automatisch einen Schwellenwert aus dem Diagramm, mit welchem die eindeutige Unterteilung in Schwarz und Weiß für die weitere Bearbeitung erfolgen kann. Dies erreicht man, in dem das Programm den niedrigsten Wert von dem Intervall zwischen ca. 400 bis ca. 750 sucht, welcher dann als eine Art Grenze zwischen den beiden Farben dient.

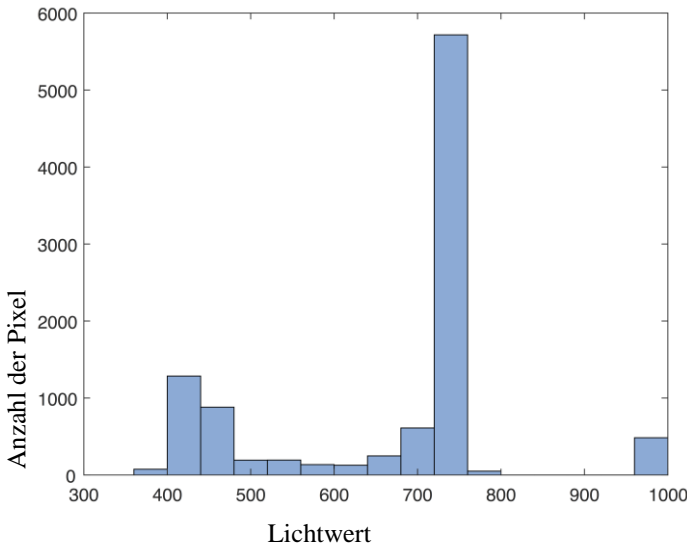


Abbildung 1: Vom Lichtsensor gemessene Werte werden können.

III. HAUPTTEIL

Nun werde ich genauer auf unsere Umsetzung eingehen. Am Anfang erwähnte ich, dass der Blatteinzug automatisch erfolgen sollte. Dies haben wir so realisiert, dass das oberste Ende des einzuziehenden Blattes unter einen Reifen geklemmt wird, welcher dann auf Knopfdruck das Blatt einzieht, außerdem wird das Blatt immer etwas weiter weingezogen, sobald der Lichtsensor eine Zeile des Blattes gescannt hat. Glücklicherweise lässt sich der Motor in Hinsicht auf Drehwinkel und Drehgeschwindigkeit ziemlich genau

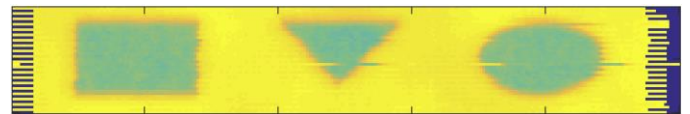


Abbildung 3: Durch Lichtwert dargestellte Formen



Abbildung 4: Formen nach Unterteilung

In den obigen Abbildungen 2 und 3 sieht man, wie das Bild vor und nach der Bearbeitung aussieht. Hier wird jedoch Weiß als Gelb und Schwarz als Blau dargestellt. Man kann zudem erkennen, dass im ersten Bild dunkelblaue Ränder vorliegen

und im zweiten nicht. Dies sind die Initialisierungswerte des Lichtsensors, wo er noch keine Messungen vollführen kann, diese wandelt das Programm sofort in Weiß um.

Nachdem diese eindeutige Unterteilung stattgefunden hat, wird jedes Objekt mit einer bestimmten Menge von zusammenhängenden schwarzen Pixeln mit einer spezifischen Zahl versehen. Durch diesen Schritt werden die einzelnen Objekte isoliert, wodurch die Formerkennung durchgeführt werden kann, außerdem werden zu kleine Objekte dadurch übersprungen, welche eventuell durch Fehlmessungen entstanden sind. Somit findet sich dann jedes Objekt ausgeschnitten auf einem einzelnen Bild vor.

Das Programm führt nun eine Faltung zur Kantenerkennung durch, dies wird mit einer in Matlab vorgegebenen Funktion durchgeführt. Hierbei wird der Laplace-Kern genutzt, welcher eine Faltungsmatrix speziell zur Kantenglättung ist. Dadurch werden nun also die Kanten des Objektes ausgeschnitten und man erhält die Hülle des Objektes wie sie in Abbildung 5 zu sehen ist.

Nun hat man alle Vorbedingungen geschaffen um nun die Flächeninhaltsberechnung und die Formerkennung durchzuführen. Hierfür werden zuerst zwei formspezifische Verhältnisse berechnet

$$V_1 = \frac{u^2}{A} \quad (1)$$

$$V_2 = \frac{A}{A_R} \quad (2)$$

Das u was in Gleichung (1) gegeben ist, entspricht der Pixelmenge der zuvor bestimmten Hülle des Objektes und das A entspricht der Pixelanzahl des ursprünglich eingescannten Objektes, das A_R welches in Gleichung (2) gegeben ist, entspricht zudem der Pixelzahl eines um die Objekte gelegtem Rechtecks. Da alle diese Werte in Pixeln angegeben sind, haben sie noch keine direkte Aussagekraft über die Größe des Objektes. Nun wird aus den berechneten Verhältnissen ein Score für jedes Objekt bestimmt, so dass nun jedes Objekt einen Score für jede Form zugewiesen hat. Also wird nach beiden Berechnungen geguckt, welcher Score jeweils am größten ist und dadurch wird letztendlich die spezifische Form des Objektes erkannt. Nun muss nur noch der Flächeninhalt bestimmt werden, dies geschieht indem der als A bestimmte Wert mit einem empirisch bestimmten Wert multipliziert wird. Welcher das Maß ist, wie viele mm^2 in einem Pixel enthalten sind. Danach hat man somit auch den Flächeninhalt gegeben, das fertige Ergebnis ist in Abbildung 5 zu sehen.



Abbildung 5: fertiges Scanergebnis

IV. ERGEBNISDISKUSSION

Wie in Abbildung 5 zu sehen, kann die Formerkennung, da dort ein Quadrat als solches erkannt wurde, und auch die Flächeninhaltsberechnung, welcher hier mit 862 mm^2 von ursprünglichen 900 mm^2 erkannt wurde, relativ verlässlich ausgeführt werden. Jedoch gibt es selbstverständlich auch hier einige mögliche Fehlerquellen.

Zum einen kann man in Abbildung 3 und 4 starke Verschiebungen bei dem Kreis und dem Dreieck sehen, welche das Ergebnis verfälschen können, ob diese durch eventuelle Messungenauigkeiten des Lichtsensors selbst kommen oder durch den Transport des Sensors auf Zahnschienen entstanden, konnten wir nicht herausfinden. Dieses Problem haben wir insofern versucht zu lösen, indem wir die Verschiebungen, sobald sie stark genug aus dem Objekt herausragen, einfach rauslöschten. Doch vor Allem bei dem Dreieck machte dies nur beschränkt Sinn, da wir das Löschen nur bei in einer Spalte vereinzelt aufgetretenen Messwerten durchgeführt haben und dies bei der Form des Dreiecks somit nicht so gut funktioniert. Außerdem konnte es bei dem Dreieck passieren, dass durch zu starke Verschiebungen die Spitze des Dreiecks abgeschnitten wurde, was auch zu Problemen bei der Formerkennung geführt hat.

Ein weiteres Problem war die Ermittlung des Schwellwertes, da diese bei starken Veränderungen des Lichteinflusses nicht unbedingt geklappt hat. So zum Beispiel, als die Sonne ungeschützt auf den Scanner eingestrahlt hat oder wenn man in einen anderen Raum mit anderer Beleuchtung gegangen ist. Unsere Lösung hierbei war das Suchen eines Tiefpunkts in einem bestimmten Intervall zwischen den beiden Maxima, aber bei zu großen Lichtveränderungen hat dies nicht zur Völle funktioniert. Eine andere Lösungsmethode wäre, dass man, wie bei handelsüblichen Druckern samt Scanner, den Sensor komplett einbaut, um ihn vor äußeren Einflüssen zu schützen, jedoch hatten wir dafür weder genug Bauteile noch genug Zeit, um solch große Konstruktionsänderung durchzuführen.

Zu guter Letzt war die Genauigkeit des Scanners an sich auch ein Problem. Um eine Genauigkeit zu erreichen um immer perfekte Ergebnisse zu erreichen, müsste der Scanner dementsprechend langsam hin- und herfahren um scharfe Messwerte zu erhalten, doch man musste einen Mittelwert zwischen Messpräzision und –geschwindigkeit finden. So hätte man wahrscheinlich für ein 3 cm hohes Objekt circa 10 Minuten gebraucht, um ein perfekt scharfes Bild zu erhalten. Da wir den Scanner jedoch vorstellen mussten, haben wir die Geschwindigkeit erhöht, weshalb also gewisse Ungenauigkeiten entstanden sind. So hätte man das Ergebnis, würde man nur auf die Akkuratess des Messergebnisses bedacht sein, so hätte man das Ergebnis durchaus noch etwas verbessern können.

V. ZUSAMMENFASSUNG UND FAZIT

Zusammenfassend kann man also sagen, dass wir in unserer Gruppe das Ziel durchaus erreicht haben. Wir waren relativ kreativ bei der Projektfindung, indem wir ein Projekt nahmen welches durchaus schon in den vorherigen Jahrgängen genutzt

wurde und haben aber noch eine individuelle Veränderung an der Arbeitsweise des Apparates vorgenommen. Die Konstruktion hätte man zudem durchaus auch noch schöner gestalten können, da wir den NXT verhältnismäßig lose an der Seite montiert haben, aber im Endeffekt geht Effizienz vor Design. Das Programm hätte man zudem auch definitiv noch effizienter und übersichtlicher schreiben können, aber da sowieso beim Programmieren eines Projektes solcher Größe nie Alles auf Anhieb funktioniert, musste man Wege finden die vielleicht nicht zeiteffizient oder übersichtlich sind, aber ihren Zweck erfüllen. So haben wir zum Schluss eine Genauigkeit von circa 60 bis 70 Prozent bei der Formerkennung und der Flächeninhalt wich maximal zu 60 mm ab, unter den gegebenen Umständen kann man mit diesem Ergebnis also durchaus zufrieden sein. Wenn man diesen Scanner für die Industrie nutzbar machen möchte, so müsste man definitiv mehr Bauteile brauchen, da er dort zum einen viel größere Objekte scannen müsste und zudem genauer Scannen müsste, was den Einsatz besserer Sensoren voraussetzen würde.

Zudem hat mich Matlab als Programmiersprache positiv überrascht, da wir sie schließlich vorher in der Uni noch nicht behandelt. Sie ist kurz gehalten und hat mit der Seite der Entwickler Mathworks eine gute Alternative um nach geeigneten Befehlen zu recherchieren. Was man jedoch etwas bemängeln kann, ist die gelegentlich nicht wirklich kontinuierliche Notation, vor Allem im Bezug darauf ob man eckige, runde oder doch mal geschweifte Klammern setzen soll. Das Projektseminar an sich, konnte ich am Ende der zwei Wochen auch mit positiven Eindruck verlassen, da es meiner Meinung nach einen geeigneten Einblick in den Studiengang der Elektro- und Informationstechnik bietet, wobei der Schwerpunkt natürlich auf dem programmieren gelegt war. So hat man die Gelegenheit gehabt wie es ist, ein eigenes Projekt zu entwerfen und dieses auch irgendwie zu realisieren. Außerdem haben wir einen Ausblick drauf gehabt, was man mit unserem aktuellen Wissensstand schon realisieren kann und außerdem was später vielleicht noch so möglich wäre. Zudem musste man auch auf das Zeitmanagement achten, was für Studenten später auch definitiv wichtig wird. So werde ich den Bericht mit diesem positiv ausfallenden Fazit in Hinsicht auf die Projektdurchführung durch die Unimitarbeiter und auch auf die Leistung unserer Gruppe in dem kleinen Zeitraum beenden.

LITERATURVERZEICHNIS

[1] Otto-von-Guericke-Universität Magdeburg | OVGU : Lego Praktikum an der Otto-von-Guericke-Universität Magdeburg | OVGU .
<https://www.youtube.com/watch?v=nVnWycZoGAs> (1:26) . Version:
21.03.2018