

Geometrie-Scanner

Leander Bartsch, ETIT
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—In dem Paper „Geometrie-Scanner“ wird die Konstruktion und Programmierung eines Scanners mithilfe von Lego Mindstorms und dem Matlab NXT Toolkit beschrieben. Der Aufbau erfolgt mit einem Lego Mindstorms NXT und Lego Standardbauteilen. Er ermöglicht das Scannen eines DIN-A4-Dokumentes, welches die einfachen geometrischen Formen Kreis, Dreieck und Quadrat enthält. Es wird erklärt, wie ein Blatt pixelweise durch einen Lichtsensor digitalisiert werden kann. Das Hauptaugenmerk liegt auf der Entwicklung einer Software, die die Verarbeitung der eingescannten geometrischen Formen mit Matlab ermöglicht. So wird die Vorverarbeitung, wie das Umwandeln eines Graustufenbildes, in ein Schwarz-/Weiß-Bild mit einem dynamischen Schwellwert beschrieben, außerdem das Nummerieren und anschließende Isolieren von Objekten. Durch das Berechnen von formspezifischen Verhältnissen wird das Erkennen der geometrischen Formen erklärt. Die Verhältnisse sind die Kompaktheit und der Flächeninhalt des umliegenden Rechtecks zum Objektflächeninhalt. Außerdem wird der Flächeninhalt der erkannten Formen berechnet.

Schlagwörter—Matlab Bildverarbeitung, Flächeninhalt, Lego Mindstorms, Objekterkennung, Scanner

I. EINLEITUNG

SCANNER dienen der Erfassung und Digitalisierung von Informationen. Überall müssen Informationen elektronisch eingelesen und verarbeitet werden. Dabei ist der Begriff Scanner nicht auf die weitverbreiteten Flachbettscanner für Dokumente beschränkt, sondern umfasst einen weiten Bereich. So gehören auch 3D-Scanner, Handscanner und Radar dazu. [1] In dieser Arbeit soll es um die Entwicklung eines Scanners für DIN-A4-Dokumente im Rahmen des Projektseminars ETIT gehen, weshalb die folgenden Ausführungen auf Dokumentenscanner beschränkt sind.

Zu einem Scanner gehört auch immer eine Software, die im einfachsten Fall das eingescannte Dokument anzeigt. Oft soll die entstandene Bild-Datei weiterverarbeitet werden. So wird Software zum Dokumentenmanagement, zur Bildverbesserung oder zur Texterkennung eingesetzt. [2] Ebenso sind ingenieurstechnische Anwendungen denkbar, wie das Einscannen von technischen Zeichnungen. Dabei kann es wünschenswert sein, dass maßstabgetreue Objekte vermessen werden. Neben Papierdokumenten können natürlich auch flache Körper zweidimensional erfasst werden.

Die Idee besteht darin, einen Scanner mit einem Lego Mindstorms NXT zu konstruieren, der ein DIN-A4-Blatt einscannen kann. Auf diesem Dokument sollen die einfachen geometrischen Formen Dreieck, Kreis und Quadrat abgebildet sein. Die Software des Scanners soll ein solches Dokument erfassen und die geometrischen Formen finden. Sie sollen voneinander isoliert und unterschieden werden. Es soll also das Erkennen der drei geometrischen Formen möglich sein.

Die Software soll außerdem den Flächeninhalt der Objekte berechnen können.

II. VORBETRACHTUNGEN

A. Scanner und Bildverarbeitungssoftware

Zum Scannen von Dokumenten sind zwei Scanner-Bauformen sehr verbreitet. Beim Flachbettscanner fährt ein lichtempfindlicher Sensor die Vorlage ab, welche selbst nicht bewegt wird. [1] Der Sensor ist dabei so breit wie das Dokument selbst. Bei einem Einzugsscanner ist es genau andersherum. Das Dokument wird eingezogen und dabei an der Sensorzeile entlangbewegt. [1] Der Nachteil ist, dass nur einzelne Seiten und keine Bücher oder ähnliches eingescannt werden können. [1]

Neben der Software zur Kommunikation mit dem Scanner existiert diverse Software zur Weiterverarbeitung von Scans. Eine wichtige Anwendung ist die Texterkennung, auch als OCR (Optical Character Recognition) bezeichnet. [3] Durch Texterkennung sollen die Buchstaben in einem Bild erkannt werden. Das ermöglicht die elektronische Weiterverarbeitung der eingescannten Texte. So ist nach einer Texterkennung das Nutzen einer Übersetzungssoftware möglich. [1] In anderen Bereichen existiert Software zur Verarbeitung von mikroskopischen Aufnahmen. [4] Sie ermöglicht das Zählen und Vermessen von Objekten. [4] Etwas ähnliches soll auch die in dieser Arbeit beschriebene Software leisten.

B. Lego-Mindstorms-Geometrie-Scanner

Der DIN-A4-Scanner soll mit Lego Mindstorms aufgebaut werden. In den Vorjahren wurden beim Lego-Projektseminar schon Scanner konstruiert. [5] Allerdings lag das Hauptaugenmerk immer auf dem Scannen selbst und nicht auf der Weiterverarbeitung der Scans.

Als optischer Sensor zur Erfassung von Graustufen steht ein Lichtsensor zur Verfügung. Dieser kann nur einen kleinen Bereich erfassen. Deshalb muss das Prinzip des Einzugs-scanners und des Flachbettscanners kombiniert werden, da nur ein einzelner Pixel, keine ganze Blattzeile erfasst werden kann. So soll, wie beim Flachbettscanner, der Sensor über das Blatt bewegt werden, um eine Zeile des Dokumentes zu scannen. Wie beim Einzugsscanner soll aber auch das Blatt bewegt werden, wodurch der Zeilenwechsel erfolgen soll. Mit diesem Vorgehen ist es möglich, die zwei nötigen Dimensionen mit dem ein Pixel großen Sensor abzutasten.

Die Software teilt sich in zwei Bereiche. Zum einen wird eine Software benötigt, die die Scanner-Hardware steuert. Außerdem ist eine Software zur Verarbeitung des Scans nötig. Die Entwicklung der benötigten Software erfolgt vollständig mit Matlab. Zur Programmierung des Lego Mindstorms NXT

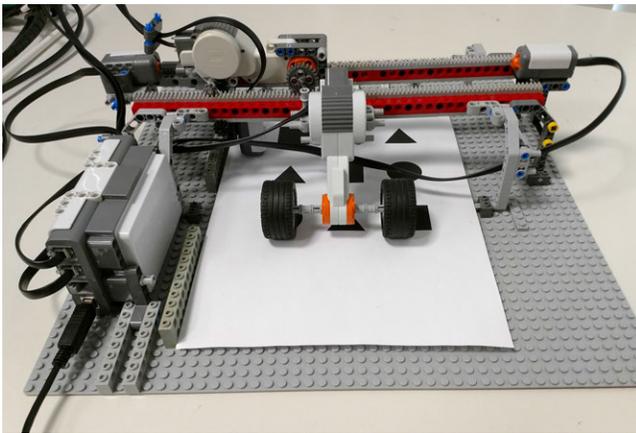


Abbildung 1. Hardware des Scanners mit eingezogenem Blatt

mit Matlab wurde die Mindstorms NXT Toolbox der RWTH Aachen genutzt.

III. HAUPTTEIL

A. Hardware des Scanners

Zunächst soll die Funktionsweise des Lichtsensors erläutert werden. Im passiven Modus wird durch eine Fotodiode die Intensität des ankommenden Lichtes gemessen. [6] Im aktiven Modus wird Licht durch eine LED ausgestrahlt. [6] Dieses Licht wird durch eine Oberfläche reflektiert und zurück auf die Fotodiode geworfen, die wiederum die Intensität bestimmt. [6] Für den Scanner wird der aktive Modus genutzt. Die Intensität wird einheitenlos als Wert zwischen 0 für ganz dunkel und 1023 für ganz hell angegeben.

Wie schon im vorherigen Abschnitt erklärt, muss der Sensor in zwei Dimensionen über das Blatt bewegt werden, da dieser nur einen einzelnen Pixel messen kann. Dazu wurde auf einer Lego-Grundplatte ein Portal gebaut. Dieses ist breiter als ein DIN-A4-Blatt, weshalb das Blatt unter dem Portal eingezogen werden kann. Dies erfolgt mit zwei motorisierten Reifen, wobei diese nur locker auf dem Blatt aufliegen, um große Kräfte zu vermeiden. Auf dem Portal fährt ein Schlitten auf Zahnschienen hin und her. Der Schlitten kann den Lichtsensor 90 Grad entgegen der Einzugsrichtung des Papiers über das Dokument bewegen. So tastet der Lichtsensor eine Zeile des Blattes nach der anderen ab. Unter dem Schlitten befindet sich der Lichtsensor zum Aufnehmen der Grauwerte. Dieser ist so tief befestigt, dass der Abstand zwischen Lichtsensor und Blatt möglichst gering ist. Auf dem Portal ist jeweils links und rechts ein Tastsensor angebracht, der als Endschalter für den Schlitten dient.

Die Abbildung 1 zeigt die vollständig aufgebaute Hardware. In der Mitte ist der Motor mit den Reifen für den Blatteinzug zu erkennen. Der links darüber liegende Motor ist der des Schlittens, welcher sich aktuell am linken Anschlag befindet.

B. Software zum Scannen eines Dokumentes

Der Scan-Vorgang lässt sich in zwei Phasen unterteilen. In der ersten Phase fährt der Schlitten über das Blatt und der Sensor zeichnet die Helligkeitsdaten auf. In der zweiten Phase



Abbildung 2. Eingesanntes Graustufenbild

wird das Blatt ein Stück eingezogen und es geht wieder mit der ersten Phase weiter.

In der ersten Phase muss zunächst der Motor zum Bewegen des Schlittens gestartet werden. Dabei soll der Schlitten sowohl bei der Fahrt von links nach rechts als auch bei der Fahrt von rechts nach links eine neue Zeile aufzeichnen. So kann auf das Zurückfahren des Schlittens verzichtet werden, was Zeit spart. Die Startposition ist immer links. Die Software muss also für den Scan einer Zeile folgende Aufgaben erfüllen: Der Motor des Schlittens muss entgegen der vorigen Richtung eingeschaltet werden. Gleichzeitig muss die Aufzeichnung der Helligkeitswerte beginnen. Die Helligkeitswerte werden mit einem konstanten Zeitabstand in einer eindimensionalen Matrix fester Größe gespeichert. Es findet keine Synchronisation der Position des Schlittens und der Messwerte statt. Deshalb muss die Fahrzeit des Schlittens möglichst konstant gehalten werden, wozu die in der NXT Toolbox verfügbare Drehzahlregelung aktiviert wird. Trotzdem schwankt die Anzahl der Messwerte von Zeile zu Zeile. Um keine Messwerte zu verlieren, muss die Matrix etwas größer als nötig gewählt werden. Dies hat zur Folge, dass einige Felder in der Matrix nicht genutzt werden. Deshalb verbleiben am Rand Initialisierungswerte, wie im Beispielscan in Abbildung 2 zu sehen. Beim Scannen von rechts nach links ist das Umdrehen und Verschieben der Pixel notwendig. Als letzter Schritt wird die eindimensionale Matrix als neue Zeile einer zweidimensionalen Matrix hinzugefügt.

Damit ist die erste Phase abgeschlossen und es folgt Phase zwei. Das Blatt wird ein Stück eingezogen, indem sich der Motor für den Einzug um einen festen Winkel dreht. Nun beginnt der Scan der nächsten Zeile in Phase eins. Die zwei Phasen können beliebig oft wiederholt werden, wobei die zweidimensionale Matrix immer größer wird. Die Breite dieser Matrix ist durch die Größe der eindimensionalen Matrix festgelegt. Hingegen ist die Zeilenanzahl beliebig. In der Software kann die Anzahl der zu scannenden Zeilen nach Bedarf eingestellt werden.

C. Erkennung der geometrischen Formen und Berechnung des Flächeninhaltes

Nachdem der Scanvorgang abgeschlossen wurde, beginnt die Auswertung. Um das Ziel, die Unterscheidung verschiedener geometrischer Formen und Ermittlung derer Flächeninhalte zu erreichen, sollen im Folgenden die nötigen Algorithmen beschrieben werden.

Die Ausgangssituation ist eine zweidimensionale Matrix, deren Felder jeweils den Graustufenwert eines Pixels enthalten. In dieser müssen die eingesannten geometrischen Formen gefunden und isoliert werden. Dieser Verarbeitungsschritt wird als Segmentierung bezeichnet. [7] Das im folgenden beschriebene Verfahren gehört zu den pixelorientierten Verfahren zur Segmentierung. [7] Pixelorientiertes Verfahren meint,

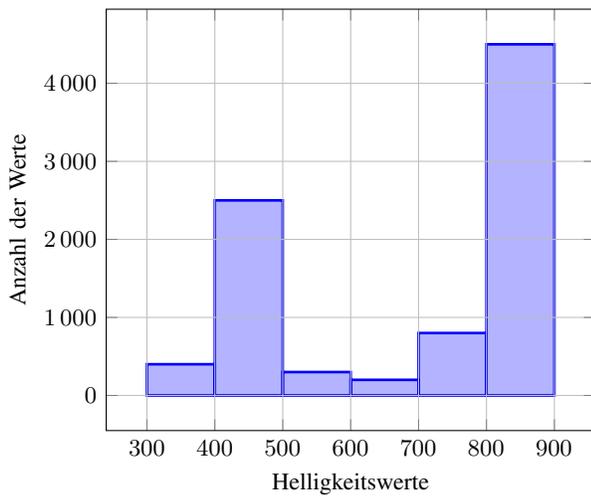


Abbildung 3. Histogramm der Helligkeitswerte (Grauwerte)

dass für jeden Pixel einzeln überprüft wird, ob dieser zum Objekt gehört oder nicht. [7] Im Fall der schwarzen Objekte auf weißem Hintergrund bietet sich eine Binarisierung an. Binarisierung bedeutet, dass jeder Pixel einem von zwei Segmenten zugeordnet wird. [8] Das eine Segment sind alle schwarzen geometrischen Formen. Das zweite Segment ist der Hintergrund. Um die Binarisierung durchzuführen, wird den Grauwerten der Pixel, je nachdem ob sie kleiner oder größer eines Schwellwertes sind, eine Eins (gehört zum Objekt) oder eine Null (gehört zum Hintergrund) zugeordnet. [7] Es wird also das Graustufenbild in ein Schwarz-/Weiß-Bild umgewandelt, was auch als Threshold bezeichnet wird. Der Schwellwert kann mit Hilfe eines Histogrammes der Grauwertverteilung gefunden werden. [7] Ein solches Histogramm ist in Abbildung 3 zu sehen und gibt Aufschluss über die Verteilung der Grauwerte. [7] Der Schwellwert findet sich in dem Histogramm in Form des Tiefpunktes beim Helligkeitswert von etwa 600 wieder. Der Hochpunkt links davon, also die Helligkeitswerte um 400, sind die Objekte. Die Grauwerte um 800 hingegen gehören zum weißen Hintergrund. Das Finden des Schwellwertes erfolgt dynamisch durch die Software, indem der Tiefpunkt im Histogramm im Bereich von 400 bis 800 gesucht wird. Das Histogramm wird aus der Scan-Matrix erstellt. Damit ist ein erster wichtiger Schritt der Bildverarbeitung abgeschlossen. Das Ergebnis ist ein Bild, welches nur noch aus weißen Hintergrundpixeln und schwarzen Objektpixeln besteht.

Das Problem ist, dass eine unbekannte Anzahl n an Objekten vorhanden ist. Im nächsten Schritt müssen also die Objekte voneinander isoliert werden. Dazu werden alle zusammenhängenden Pixel, die zu einem Objekt gehören, mit einer Nummer versehen. Diesen Schritt nennt man Labeln der Objekte und ist in Matlab als Funktion vorhanden. [9] Es werden die Einsen der Pixel durch die Nummer n des Objektes ersetzt. Die Hintergrundpixel bleiben Null aber die Pixelwerte der Objekte sind nun 1 bis n , je nachdem zu welchem Objekt sie gehören.

Alle folgenden Schritte werden für alle 1 bis n Objekte einzeln durchgeführt. Zunächst werden alle bis auf ein Objekt



Abbildung 4. Dreieck nach der Faltung mit dem Laplace-Kern

aus dem Bild gelöscht. Daraufhin wird durch Zählen der Pixel des verbleibenden Objektes der Flächeninhalt A_O ermittelt. Das Objekt wird übersprungen, wenn das Objekt weniger als 50 Pixel groß ist (wenn $A_O < 50$). So werden Fragmente, die kein Objekt sein können, herausgefiltert.

Es folgt die Ermittlung des Flächeninhaltes des umliegenden Rechteckes A_R . Dazu werden alle Zeilen und Spalten der Matrix, die nur Nullen enthalten, gelöscht. In diesem Schritt werden auch Einzelpixel abgeschnitten, die als einzige außerhalb des Objektes liegen und damit wahrscheinlich durch einen Zeilenversatz entstanden sind. Es bleibt eine kleinere Matrix übrig, deren Größe der Flächeninhalt A_R ist und deren Breite und Höhe der maximalen Breite und Höhe des Objektes entspricht.

Nun soll durch eine lineare Faltung die Kante isoliert und anschließend der Umfang u des Objektes ermittelt werden. Eine Faltung ist eine mathematische Operation, für die ein Bild und ein Faltungskern benötigt wird. [10] Sowohl das Bild als auch der Faltungskern ist eine Matrix. [10] Durch Verrechnen beider Matrizen entsteht ein Ergebnisbild, welches eine Veränderung des Bildes je nach Art des Faltungskernes darstellt. [10] Wenn als Faltungskern der Laplace-Kern genutzt wird, bleibt im Ergebnisbild nur die Kante des Objektes über. Es bleiben nur die Pixel schwarz, welche an einem Farbübergang im Ausgangsbild, also an einer Kante, liegen. [10] Die Ermittlung des Umfangs u erfolgt, indem die Anzahl der verbliebenen Pixel ermittelt wird. Es werden also die Konturpixel des Objektes gezählt. Die Abbildung 4 zeigt ein ausgeschnittenes Dreieck nach der Faltung mit dem Laplace-Kern.

Damit ist die Vorverarbeitung zur Identifizierung eines Objektes abgeschlossen. Die Identifizierung erfolgt nun durch die Berechnung von zwei Verhältnissen; die jeweils spezifisch für eine bestimmte geometrische Form sind. Das erste Verhältnis V_1 (vgl. (1)) ergibt sich aus dem Quadrat des Objektumfanges u^2 zu dem Flächeninhalt des Objektes A_O .

$$V_1 = \frac{u^2}{A_O} \tag{1}$$

Dieses Verhältnis wird Kompaktheit genannt und ermöglicht die Unterscheidung von Objekten. [7] Das Verhältnis ergibt für symmetrische, geometrische Formen, wie Kreis, Quadrat oder gleichseitiges Dreieck einen eindeutigen Wert.

Das Verhältnis V_2 (vgl. (2)) ist der Flächeninhalt vom Objekt A_O zum Flächeninhalt des umliegenden Rechtecks A_R .

$$V_2 = \frac{A_O}{A_R} \tag{2}$$

Dieses Verhältnis ist nicht so eindeutig wie V_1 . Dennoch ist es notwendig, da die Bestimmung des Umfangs u nicht genau genug ist und damit die Toleranzen für V_1 zu groß sind.

Die eigentliche Identifizierung erfolgt nun durch ein Punktesystem. In der Software sind Wertebereiche für die Verhältnisse V_1 und V_2 für die drei zu identifizierenden Objekte hinterlegt. Für jedes mögliche Objekt, also Kreis, Quadrat und möglichst gleichseitiges Dreieck, wird ein Score berechnet. Das mögliche Objekt mit dem höchsten Score ist die vorliegende geometrische Form. Je schlechter der Scan ist, desto kleiner und damit uneindeutiger sind die Scores.

Als letzter Schritt erfolgt die Berechnung der Flächeninhalte. Dazu wird der Flächeninhalt A_O , also die Pixelanzahl, mit einem Faktor multipliziert. Dieser Faktor skaliert die Pixelanzahl des Bildes auf die reale Objektgröße. Je nachdem, als welche geometrische Form das Objekt klassifiziert wurde, wird der Faktor leicht variiert. Dies ist notwendig, da aufgrund des Rauschens an den Objektkanten der Flächeninhalt A_O formabhängig schwankt.

IV. ERGEBNISDISKUSSION

Die an den Scanner gestellten Anforderungen werden fast vollständig erfüllt. So ist es möglich, ein DIN-A4-Blatt einzuscannen. Auch funktioniert die Weiterverarbeitung der eingescannten geometrischen Formen. Die Software ermöglicht die Erkennung der drei Objektarten Kreis, Quadrat und Dreieck. Allerdings funktioniert es nur, wenn das Quadrat und das Dreieck mit einer Seite parallel zur Bewegungsrichtung des Schlittens liegen. Diese Bedingung resultiert daraus, dass nur so das Objekt richtig ausgeschnitten und damit der Flächeninhalt des umliegenden Rechtecks berechnet werden kann. Außerdem ist die Erkennung von Dreiecken auf fast gleichseitige Dreiecke beschränkt.

Im Folgenden sollen Probleme bei der Konstruktion der Hardware erläutert werden. Zum Bau des Portals waren einige Versuche notwendig, um es stabil genug zu bauen. Das Konstruktionskonzept mit den Zahnschienen setzt voraus, dass sich der Motor direkt am Schlitten befindet. Deshalb muss neben dem Kabel des Lichtsensors auch das Kabel des Motors immer mit bewegt werden. Dies stellt ein Problem dar, weil die Lego Mindstorms Kabel sehr steif sind und so einen hohen Zug auf den Schlitten ausüben. Das Problem konnte reduziert werden, indem eine Kabelführung gebaut wurde.

Der räumliche Abstand zwischen den Messwerten in einer Zeile und der Vorschub zwischen den Zeilen ist verschieden groß. So werden, wie in Abbildung 2 zu sehen, alle Objekte verzerrt. Sie erscheinen flacher als sie wirklich sind. Dieses Problem konnte nur bedingt gelöst werden, da der Vorschub zwischen den Zeilen nicht beliebig klein wählbar ist. Auch konnte der Abstand der Messwerte in einer Zeile nicht einfach vergrößert werden, da so die Qualität des Scans stark abnahm.

Das wohl größte Problem war der Versatz zwischen den eingescannten Zeilen. Dieser resultiert daraus, dass der Datensatz aus dem Lichtsensor und die Position des Schlittens nicht miteinander synchronisiert werden. Das Problem wurde nicht vollständig gelöst, konnte aber gut genug reduziert werden. Zum einen werden die Zeilen um einen experimentell ermittelten Wert gegeneinander verschoben. Dieses Verschieben gleicht einen systematischen Fehler beim Scannen von der linken und rechten Richtung aus. Zum Anderen werden einzelne stark

verschobene Zeilen von den Objekten abgeschnitten, um die Flächeninhaltsberechnung des umliegenden Rechtecks nicht zu verfälschen. So eine einzelne stark verschobene Zeile ist in der Abbildung 4 zu sehen.

Ein weiteres Problem war die Verschiebung des Grauwertbereiches durch unterschiedliches Umgebungslicht. Daraus resultierte ein anderer Schwellwert für die Binarisierung. Das Problem wurde durch das dynamische Ermitteln des Tiefpunktes im Histogramm gelöst.

Theoretisch reicht die Kompaktheit (vgl. (1)) aus, um einfache geometrische Formen zu unterscheiden. Das Berechnen des Umfanges ist sehr ungenau, da dieser nur durch Zählen der Randpixel ermittelt wird. Daraus resultiert auch eine ungenaue Kompaktheit, weshalb es nötig war, das Verhältnis V_2 einzuführen. Leider war es damit nicht mehr möglich, schief liegende geometrische Formen zu unterscheiden, da bei diesen die Berechnung des Flächeninhaltes des umliegenden Rechtecks sehr komplex wird.

V. ZUSAMMENFASSUNG UND FAZIT

Das Paper beschreibt den Bau eines Scanners für DIN-A4-Dokumente mit Lego Mindstorms und einer Matlab-Software zur Bildverarbeitung. Die beschriebene Konstruktion ermöglicht die Erfassung eines DIN-A4-Blattes. Dazu wurde in der Arbeit zum Projektseminar ETIT ein Verfahren zur Segmentierung von Formen erklärt. Außerdem wurde ein sehr einfaches Verfahren zum Unterscheiden der Objekte auf Basis zweier Verhältnisse aus geometrischen Größen beschrieben. Dieses funktioniert zwar prinzipiell, ist aber recht unzuverlässig. Das Ergebnis verbessert sich mit zunehmender Größe der Objekte bzw. zunehmender Auflösung der Scans, da die Kanten sauberer werden. So ließe sich das Problem lösen, indem die Qualität der Scans verbessert wird oder indem durch eine verbesserte Vorverarbeitung die Kanten sauberer werden. Außerdem wäre es wünschenswert, wenn weitere und beliebig liegende Objekte erkennbar wären. Dazu müsste das Berechnen des umliegenden Rechtecks für beliebige Objekte ermöglicht werden. Trotz alledem ist das Endergebnis sehr zufriedenstellend.

LITERATURVERZEICHNIS

- [1] WIKIPEDIA, THE FREE ENCYCLOPEDIA: *Scanner (Datenerfassung)*. [https://de.wikipedia.org/wiki/Scanner_\(Datenerfassung\)](https://de.wikipedia.org/wiki/Scanner_(Datenerfassung)). Version: 12. Januar 2018
- [2] WIKIPEDIA, THE FREE ENCYCLOPEDIA: *Dokumentenscanner*. <https://de.wikipedia.org/wiki/Dokumentenscanner>. Version: 12. Januar 2018
- [3] UBNTUSERS: *Texterkennung*. <https://wiki.ubuntuusers.de/Texterkennung/>. Version: 16. März 2018
- [4] GSA GMBH: *GSA Image Analyser*. <https://image.analyser.gsa-online.de/?lang=de>. Version: 16. März 2018
- [5] OTTO-VON-GUERICKE-UNIVERSITÄT MAGDEBURG: *Lego-Praktikum an der Otto-von-Guericke-Universität Magdeburg — OVGU*. <https://www.youtube.com/watch?v=nVnWycZoGAs>. Version: 18. März 2018
- [6] BANGEMANN, Felix: *Lego Mindstorms in der FEIT*. Otto von Guericke Universität Magdeburg Fakultät für Elektrotechnik und Informationstechnik, 2018
- [7] MEYER, Michael: *Segmentierung*. <http://www.weblearn.hs-bremen.de/risse/AWI/SEGMENT/SEGMENT1.HTM>, 8. März 2018
- [8] HENDRIK HORN, Jörgen K.: *Segmentierung von Bilddaten*. <http://mneenth.de/static/data/Segmentierung.pdf>, 2004
- [9] XIAOLU ZHAO, JUANLU: *Labeling mit MATLAB*. <http://pille.iwr.uni-heidelberg.de/~ocr02/Labeling%20mit%20MATLAB.html>. Version: 13. März 2018
- [10] RN-WISSEN: *Bildverarbeitung Tutorial*. http://rn-wissen.de/wiki/index.php?title=Bildverarbeitung_Tutorial. Version: 16. März 2018