

Sally can Draw: Automatisierter Light-Painting-Roboter

Michelle Horn, Computervisualistik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Im Rahmen des Projektseminars Elektrotechnik/ Informationstechnik 2022 wurde ein Light-Painting-Roboter entwickelt. Dieser ist in der Lage, Kanten in einem Bild zu finden und diese mit einer Leuchtquelle, vor einem Hintergrund, nachzuzeichnen. Im Folgenden werden verschiedene Konstruktionsmöglichkeiten, sowie ihre Vor- und Nachteile beleuchtet. Außerdem werden aufgetretene Probleme und Lösungen erläutert. Das mit MATLAB implementierte Programm wird in den Schritten Bildverarbeitung und Kantenverfolgung erklärt.

Schlagwörter—Bildverarbeitung, LEGO-Mindstorms, Light-Painting, MATLAB, Roboter, Segmentierung

I. EINLEITUNG

LIGHT-PAINTING, oder auch Light-Writing, ist eine fotografische Technik, die mittels Langzeitbelichtung durchgeführt wird. Dabei wird im dunklen mit einer oder mehreren Lichtquellen gemalt oder geschrieben. Besonders schöne Effekte entstehen, wenn Light-Painting vor einem Hintergrund angewendet wird. Die Ergebnisse sind erst später auf dem Ergebnisbild sichtbar. Ist eine Linie gezeichnet kann sie nicht wieder entfernt werden, was bedeutet, dass ein Fotograf in der Regel viele Versuche benötigt, um ein Motiv wie gewünscht zu fotografieren. Um die Gesamtdauer eines Light-Painting-Bildes zu verkürzen, wurde *Sally can Draw*, der automatisierte Light-Painting-Roboter, konzipiert. So soll es durch eine exakte Linienführung möglich sein, das ideale Ergebnisbild in wenigen Versuchen aufzunehmen. Des Weiteren ist es so nicht notwendig, selbst zu zeichnen, was das Wechseln zwischen der Kamerabedienung und dem Zeichnen erspart. So ist es potenziell möglich, die Belichtungszeit zu reduzieren. Ab einer bestimmten Belichtungszeit kann das Bild bzw. die Umgebung auf dem Bild ansonsten nämlich zu hell werden. Der Roboter soll möglichst genau zeichnen. Dabei soll die Lichtquelle verdunkelt oder deaktiviert werden können, um keine fehlerhaften Linien zu erzeugen, wenn das Bild aus mehr als einem geschlossenen Kantenzug besteht. Außerdem wird das Zeichnen vor einer 3D-Szene als Hintergrund bevorzugt, weshalb der Light-Painting-Roboter in der Lage sein muss, im Raum zu zeichnen. Weiterhin werden Anforderungen an das Programm gestellt. Es muss in der Lage sein, eingelesene Bilder zu verarbeiten und darin Kantenzüge zu segmentieren. Dabei soll die Berechnungsdauer gering sein und es dürfen nicht zu viele Befehle in kurzer Zeit an den NXT-Stein übertragen werden.

II. VORBETRACHTUNGEN

Es ergeben sich drei Konstruktionsansätze zur Umsetzung des Light-Painting-Roboters. Dabei ist die in Abschnitt II-C beschriebene Variante den anderen vorzuziehen, weil neben voller Funktionalität auch die Anforderung, vor einem Hintergrund zu zeichnen, erfüllt ist.

A. Umsetzung als Arm mittels inverser Kinematik

Der Ansatz, den Light-Painting-Roboter durch einen Arm mittels inverser Kinematik umzusetzen, wird zuerst verfolgt. Wie schematisch in Abbildung 1 zu sehen ist, hat der Arm zwei Freiheitsgrade und sein Endeffektor ist eine abdeckbare Kiste. Diese bietet Platz für eine AA-Batterie und eine LED. Hierbei werden drei Motoren verwendet, zwei zum Steuern des Armes und einer, um die Kiste mit einer Klappe abzudecken. Der NXT-Baustein befindet sich am Standsockel des Armes und bildet damit gleichzeitig ein Gegengewicht. Durch die limitierte Kabellänge ist der Arm sehr klein. Das beschränkt das Zeichnen auf einen kleinen Radius. Der erreichbare Detailgrad beim Zeichnen ist somit fraglich. Durch das Eigengewicht der Motoren kommt es zum Absacken der Armteile. Außerdem gibt es Stabilitätsprobleme. Wegen diesen Problemen wird dieser Ansatz nicht weiter verfolgt, obwohl er den Vorteil bietet, vor einem Hintergrund zeichnen zu können.

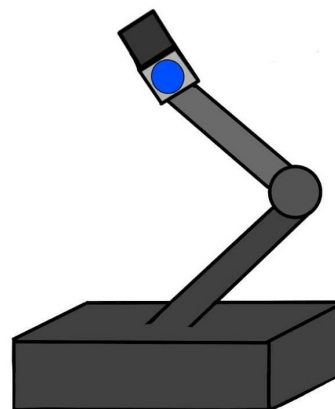


Abbildung 1. Skizzenhafter Entwurf des Light-Painting-Roboterarms

B. Umsetzung als Fahrzeug

Die Umsetzung als Fahrzeug löst die Stabilitätsprobleme, sowie die Limitierung durch die Kabellänge. Wie in Abbildung 2 zu sehen, ist der Prototyp frontgetrieben und verfügt

über die gleiche Klappvorrichtung wie in dem Abschnitt II-A beschriebenen Ansatz. Die zwei verbleibenden Motoren bilden die Vorderachse. Dabei ist ein Motor zum Vorwärts- und Rückwärtsfahren und der andere zum Lenken. Die Umsetzung als Fahrzeug erlaubt es lediglich auf dem Boden – also einem schwarzen Hintergrund zu zeichnen. Dafür ist es notwendig, die Kamera senkrecht über dem Boden zu platzieren, was zu Problemen führt. Zum einen ist ein Fernauslöser notwendig. Des Weiteren ist die Platzierung nicht mit einem Standardstativ möglich.



Abbildung 2. Prototyp des Fahrzeuges

C. Umsetzung als Gerüstkonstruktion

Die Gerüstkonstruktion besteht aus einem Rahmen an dem, wie in Abbildung 3 zu sehen ist, oben der NXT-Baustein sowie zwei Motoren angebracht sind. Der genaue technische Aufbau ist im Abschnitt III beschrieben. Als Leuchtmittel ist hierbei der in Abschnitt II-D erklärte Calliope mini verbaut. Die Umsetzung mit dem Gerüst vermeidet ebenfalls Stabilitätsprobleme und eine Limitierung durch die Kabellänge. Zudem ermöglicht sie das Zeichnen vor einem Hintergrund und bietet einen großen Zeichenradius. Das Fotografieren mit einem Standardstativ und ohne Fernauslöser ist gut umsetzbar.

D. Leuchtmittel

Eine einfache LED, die mit einer AA-Batterie betrieben wird, reicht als Leuchtmittel zum Zeichnen bereits aus. Sie ist nicht steuerbar und somit entweder an oder aus. Da eine Ansteuerung über den NXT nicht möglich ist, wird die LED vor dem Starten der Aufnahme manuell angeschaltet und dann mit einer motorisierten Klappe verdunkelt, um keine falschen Linien im Ergebnisbild zu erzeugen.

Der Calliope mini ist ein zu Bildungszwecken entwickelter Mikrocontroller [1]. Im Gegensatz zur einfachen LED kann die LED auf dem Calliope in verschiedenen Farben leuchten. Der Calliope mini ist einfach programmierbar, dazu sind verschiedene webbasierte Entwicklungsumgebungen nutzbar. Durch die Verwendung des Calliope mini kann auf die Nutzung der motorisierten Klappe verzichtet werden. Dazu werden in einem Testdurchlauf die Zeiten gemessen, in denen die Lichtquelle an und aus sein muss. Anschließend wird mittels dieser Zeiten ein Programm geschrieben, auf den Calliope mini geladen und parallel zur Kameraaufnahme gestartet.

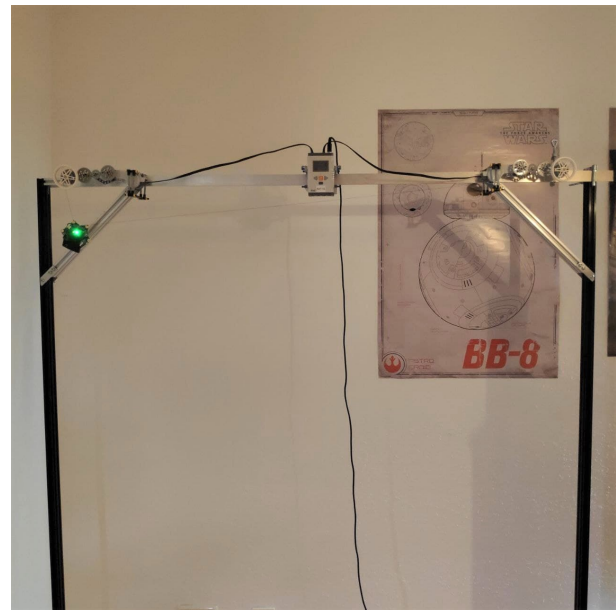


Abbildung 3. Gerüstkonstruktion mit montiertem Calliope mini

III. KONSTRUKTION

Die finale Umsetzung ist die Gerüstkonstruktion. Wie bereits im Abschnitt II-C erwähnt, besteht diese Konstruktion aus einem Metallrahmen. Auf der oberen Metallschiene sind der NXT-Baustein und zwei Motoren durch LEGO angebracht. Um den Zeichenradius zu maximieren sind die Motoren über eine Zahnradkonstruktion, wie in Abbildung 4 zu sehen, verlängert. Am Ende befindet sich jeweils eine Winde, auf die Fäden aufgewickelt ist. An diesen Fäden ist der Calliope mini befestigt. Hier sind alle leuchtenden Bestandteile außer der RGB-LED abgeklebt, um doppelt gemalte Linien im Ergebnisbild zu verhindern. Die Fäden werden durch die Motoren auf- und abgewickelt, um die Lichtquelle entlang der Kanten im Bild zu bewegen. Um ein Abspringen der Fäden von den Winden zu unterbinden, ist vor den Winden ein zusätzliches Rad angebracht.

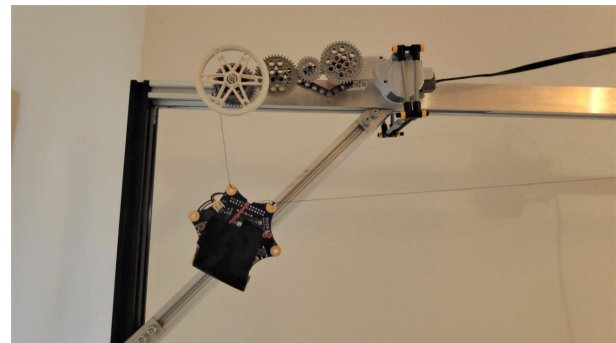


Abbildung 4. Verlängerung der Motoren und modifizierter Calliope mini

IV. SOFTWARE

Die softwareseitige Umsetzung erfolgt in zwei Schritten, der Bildverarbeitung und dem Algorithmus zum Bewegen

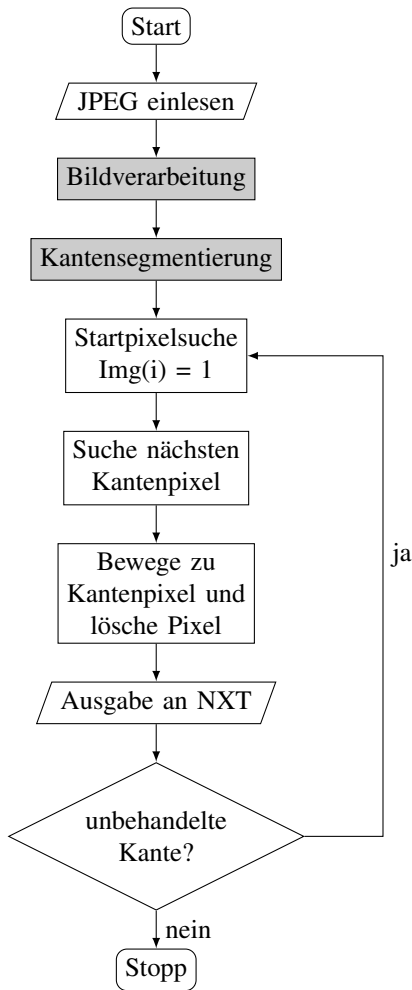


Abbildung 5. Programmablaufplan zur Erklärung des Programms

der Lichtquelle. Dazu gehört auch die Ansteuerung des NXT-Bausteins. Der Programmablaufplan in Abbildung 5 verdeutlicht die Umsetzung.

A. Bildverarbeitung

Bei der softwareseitigen Umsetzung wird zunächst ein Bild eingelesen und in ein Grauwertbild umgewandelt. Dieses Bild wird an die Funktion `highpass.m` übergeben. Diese führt auf dem übergebenen Bild eine Filterung mittels des Laplacian-of-Gaussian-Filters durch. Dieser entspricht einer Rauschreduktion mit anschließender Hochpassfilterung. Dabei ist es gegebenenfalls notwendig, den Parameter σ manuell anzupassen, sollte das Ergebnis nicht ausreichend für die Weiterverarbeitung sein. σ bestimmt dabei die Breite der Kanten. Je kleiner σ ist desto breiter sind die Kanten. Ein größer gewähltes σ führt zu schmaleren oder keinen Kanten. Der für die Beispielbilder gewählte Wert liegt zwischen 0,3 und 0,6. Das resultierende Kantenbild ist in Abbildung 6 (mitte) zu sehen. σ liegt hier bei 0,4.

Dieses Kantenbild wird an die Funktion `GetEdges.m` übergeben. Hier werden die Kanten mit dem Canny-Edge-Operator segmentiert. Dabei entsteht ein Binärbild. Eins steht für Orte an denen eine Kante ist, sonst steht dort eine Null.

Danach werden zu kleine Kanten entfernt. Das ist notwendig, da meist kleine fälschlicherweise erkannte Kanten in der Nähe von echten Kanten auftreten. Dabei werden Kantenzüge gelöscht, die aus weniger Pixeln als ein Schwellenwert bestehen. Dieser Schwellenwert wird manuell angepasst und ist stark vom jeweiligen Bild und auch vom gewählten σ abhängig. Für die getesteten Beispielbilder liegt dieser Wert zwischen 50 und 500. Das Ergebnis dieses Schrittes ist in Abbildung 6 (unten) zu sehen. Die Schwelle zum Löschen kleiner Kanten ist hier auf 400 gesetzt.

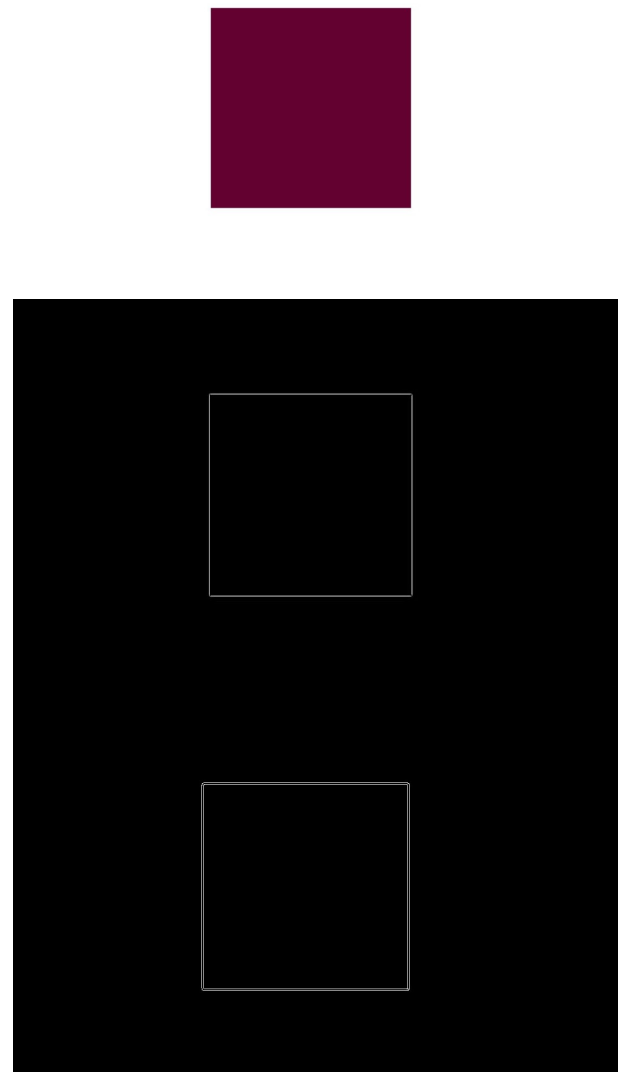


Abbildung 6. Originalbild (oben), Kantenbild als Resultat der Filterung (mitte), segmentierte Kanten (unten)

B. Kantenverfolgung

Als Ausgangsposition für den Calliope mini ist die obere linke Ecke des Rahmens definiert. Auf dem Binärbild wird das Startpixel gesucht, indem das Bild, ausgehend von der linken oberen Ecke, traversiert wird. Das Startpixel ist hierbei das erste Einspixel. Von diesem Pixel werden die umliegenden Pixel in

der 8-Nachbarschaft betrachtet. Ist eines dieser Pixel ebenfalls Teil der Kante wird das aktuelle Pixel gelöscht und das gefundene Pixel wird betrachtet. Wird kein nächstes Kantenpixel gefunden, wird in einem größeren Umkreis erneut gesucht. So werden kleine Löcher im Kantenzug überbrückt. Dieser Radius wird schrittweise größer, um bei einer abgeschlossenen Kante einen neuen Startpixel der nächsten Kante zu finden. Wird bei diesem Prozess keine Kante gefunden, wurde das gesamte Bild gezeichnet und der Algorithmus terminiert. Über die Position der Pixel im Bild wird der Abstand zu den Motoren berechnet. Daraus folgt die benötigte Fadenlänge. Hierzu werden die Motoren pro Pixel um 3° bewegt.

V. ERGEBNISDISKUSSION

In Abbildung 7 ist das aus dem Originalbild in Abbildung 6 (oben) resultierende Ergebnisbild zu sehen. Die Belichtungszeit dieses Fotos liegt bei 18 Sekunden, es wurde mit Blende 29 und ISO-100 fotografiert. Es ist also möglich, komplexere Bilder zeichnen zu lassen, ohne dass die Bildqualität durch Umgebungslicht beeinflusst wird. Dabei fällt auf, dass die Linienführung nicht optimal ist. Horizontale Linien biegen sich leicht nach unten und senkrechte Linien verlaufen nur teilweise gerade. Nachdem die Linie größtenteils gerade verläuft macht sie einen Knick um die Position zu korrigieren. Bei geschlossenen Kantenzügen erreicht die Linie ihren Anfangspunkt erneut genau.

Zum Problem der ungenauen Linienführung ist bisher keine Lösung bekannt. Lösbar sind jedoch einige Konstruktionsprobleme. Aufgrund der verschiedenen Ansätze, um den Light-Painting-Roboter zu konstruieren, die in Abschnitt II aufgeführt sind, ist die entstandene Konstruktion als Prototyp zu betrachten. Trotzdem löst dieser Ansatz viele Probleme aus den vorherigen Entwicklungsstufen. Die Stabilität ist weitestgehend gewährleistet. Es gibt minimale Bewegungen der gesamten Konstruktion bei Bewegung der Motoren. Diese hängen mit der Inkompatibilität zwischen dem Gerüst und LEGO zusammen. Die Limitierung des Zeichenradius durch die begrenzte Kabellänge wird durch die im Abschnitt III erläuterte Erweiterung behoben. Zudem ist das Zeichnen vor einem Hintergrund mit optimaler Perspektive möglich. Daraus folgt auch eine einfache Handhabung beim Fotografieren mittels eines standardmäßigen Stativs.

VI. ZUSAMMENFASSUNG UND FAZIT

Der als beweglicher Arm, Fahrzeug und Gerüstkonstruktion konzipierte Light-Painting-Roboter wird als Gerüstkonstruktion umgesetzt. Dabei werden sowohl die technische als auch die softwareseitige Umsetzung veranschaulicht. Außerdem werden aufgetretene Probleme, ihre Lösungen und die Ergebnisse dargestellt. Der Prototyp ist in der Lage einfache Kantenzüge nachzuzeichnen, bietet aber Potential zur Verbesserung.

Der Light-Painting-Roboter kann durch ein mobileres und stabileres Gerüst ergänzt werden. Ein ansprechenderes Design kann erarbeitet werden. Außerdem können auf der Softwareseite einige Verbesserungen bzw. Erweiterungen vorgenommen werden. Zum einen sollte das Problem der Linienführung gelöst werden. Zudem kann der Algorithmus

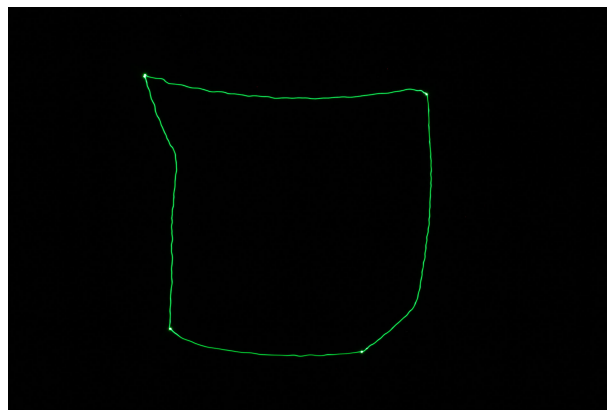


Abbildung 7. Ergebnisbild

um eine Tiefensuche ergänzt werden, um die Reihenfolge, in der die Kanten abgefahren werden, zu beeinflussen. Die Programmierung des Calliope mini kann über eine Bluetooth Verbindung durchgeführt werden, ohne den Calliope mini bei Fertigstellung eines neuen Programms mit dem Computer verbinden zu müssen. Dabei kann über eine Einbindung der Programmierung in MATLAB nachgedacht werden. Die Einstellung der Parameter in der Bildverarbeitung kann beispielsweise über eine GUI nutzerfreundlicher gestaltet werden.

Zusammengefasst bietet der Light-Painting-Roboter viel Potential zur Verbesserung und Erweiterung.

LITERATURVERZEICHNIS

- [1] CALLIOPE GGMBH: *Calliope mini, technische Daten*. <https://calliope.cc/calliope-mini/technische-daten>. Version: Februar 2022