

Farbsortiermaschine

Sebastian Knospe, Elektro- und Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Während des LEGO-Praktikums erhielten die Studierenden nicht nur eine theoretische Einführung in die Programmierung mit MATLAB, einer Software zur grafischen Darstellung von Ergebnissen und zur Lösung mathematischer Probleme, sondern sie konnten auch ihrer Kreativität beim praktischen Bau eigener LEGO-Roboter freien Lauf lassen. Im Verlauf des zweiwöchigen Praktikums wurde eine Maschine entwickelt, die mit Hilfe von LEGO Mindstorms Sensoren und Motoren in der Lage ist, Objekte nach ihrer Farbe zu sortieren.

Schlagerwörter—Farberkennung, LEGO Mindstorms, MATLAB, Programmierung, Sortiermaschine

I. EINLEITUNG

DER stetige technologische Fortschritt hat zu einer kontinuierlichen Steigerung der Produktionsgeschwindigkeiten in der industriellen Fertigung geführt.

Diese Entwicklung erfordert eine gleichzeitige Anpassung der Qualitätssicherungsmaßnahmen, um den höchsten Standards und Anforderungen gerecht zu werden und den hohen Geschwindigkeiten zu entsprechen.

In vielen Fällen reicht eine manuelle Prüfung der Produkte nicht mehr aus, um den hohen Taktzahlen der Fertigungsprozesse gerecht zu werden. Die Qualitätskontrolle muss daher zunehmend von automatisierten Systemen wie Robotern übernommen werden, die aufgrund ihrer herausragenden Eigenschaften – wie unvergleichliche Präzision und außerordentliche Schnelligkeit – in der Lage sind, die Anforderungen an eine präzise und effiziente Qualitätssicherung zu erfüllen.

Aus einer außerordentlichen Leidenschaft für das Qualitätsmanagement entstand die Idee, die Sortierung von Objekten auf intelligente Weise effizienter zu gestalten. Diese Idee beinhaltet die Entwicklung einer Sortiermaschine, die in der Lage ist, Objekte anhand ihrer Farbe zu erkennen und gezielt zu sortieren.

II. VORBETRACHTUNGEN

Im folgenden Abschnitt wird die Kernidee des Konzeptes im Hinblick auf die Schaffung eines klaren Verständnisses für die Umsetzung erläutert. Insbesondere werden die für die Realisierung der Farbsortiermaschine notwendigen Komponenten vorgestellt.

A. Was soll die Sortiermaschine alles können?

Die Farbsortiermaschine hat die Aufgabe, Objekte nach ihrer Farbe in die dafür vorgesehene Farbkiste zu sortieren. Dazu wird das Objekt auf ein Förderband gelegt, welches nach Betätigung eines Tasters in Bewegung gesetzt wird. Mit einer Kamera wird dann ein Foto des Objektes erstellt, aus dem die

Farbe des Objektes extrahiert wird. Nach der Ermittlung der Farbe werden die Sortierkisten so positioniert, dass das Objekt entsprechend seiner Farbe in die dazugehörige Kiste fällt.

B. Welche Komponenten werden dafür benötigt?

Für die Realisierung steht ein LEGO-Mindstorms-Set zur Verfügung, welches eine Vielzahl verschiedener Elemente enthält. Diese sind in Abbildung 1 dargestellt, darunter verschiedene Sensoren, Motoren, LEGO-Bausteine und vieles mehr. Der NXT-Stein fungiert als unsere zentrale Steuereinheit, an der sämtliche Motoren, Sensoren und schließlich auch der Laptop, auf dem das Programm läuft, angeschlossen sind. Ein Motor wird verwendet, um das Förderband anzutreiben, und ein weiterer wird verwendet, um die Kisten für die Sortierung zu bewegen. Ein Ultraschallsensor und ein Tastsensor dienen zur genauen Positionsbestimmung der Sortierkisten, während ein zweiter Tastsensor zum Starten des Förderbandes verwendet wird. Zudem wird eine Kamera zur Farberkennung eingesetzt.

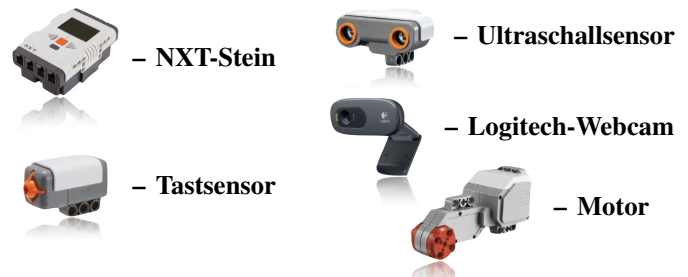


Abbildung 1. LEGO-Mindstorms-Komponenten entnommen aus [3]–[7]

III. UMSETZUNG

In diesem Abschnitt wird die Umsetzung der Farbsortiermaschine in einzelnen Schritten erläutert, beginnend mit einer detaillierten Funktionsdarstellung und dem Aufbau der Maschine. Dieser beinhaltet eine Funktionsbeschreibung der Komponenten. Abschließend wird auf die Programmierung im Hinblick auf die Arbeit mit einer Kamera eingegangen.

A. Funktionsweise

Die Funktionsweise der Sortiermaschine ist im folgendem Programmablaufplan dargestellt, der in Abbildung 2 veranschaulicht wird.

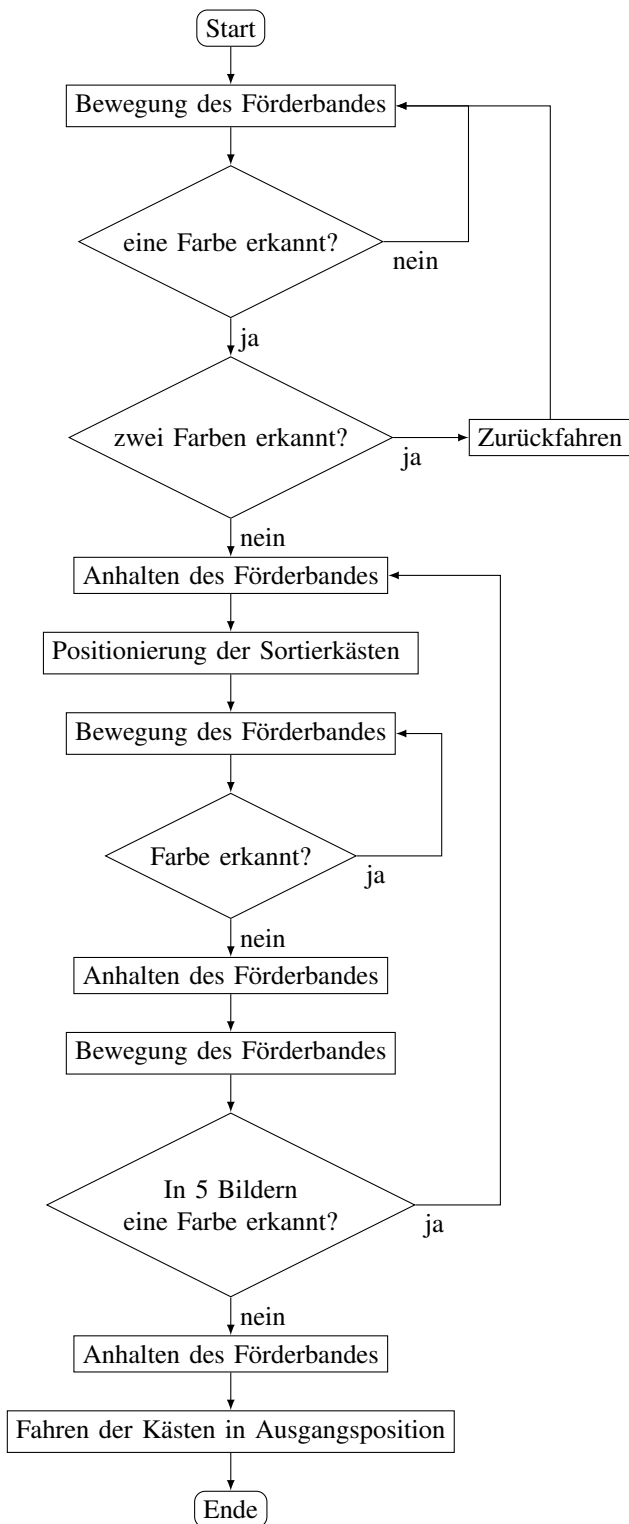


Abbildung 2. Programmablaufplan des Sortiervorganges

B. Aufbau und Funktion der Komponenten

Die Sortiermaschine in der Abbildung 3 besteht aus 3 Hauptkomponenten, dem Förderband, der Kamera und den beweglichen Sortierkästen. Das Förderband dient zum Transport der farbigen Objekte und besteht aus zwei hintereinander geschalteten Förderbändern. Diese beiden Bänder sind über

ein einfaches Getriebe miteinander verbunden, das aus drei Zahnrädern, zwei Wellen (einer Antriebs- und einer Abtriebswelle) und einer zentralen Achse zur Verbindung der Komponenten besteht. Das Förderband wird von einem Motor angetrieben, der erst nach Betätigung eines Tasters in Gang gesetzt wird.

Die Farberkennung der auf dem Förderband transportierten Objekte erfolgt durch eine Logitech Webcam, die in regelmäßigen Abständen Bilder aufnimmt. Diese Kamera ist extern mit einem Laptop verbunden und wird über MATLAB, einer Software zur Datenanalyse und -verarbeitung, in den Arbeitsablauf integriert. Obwohl auch ein Farbsensor eine Option gewesen wäre, hätte dies die Objekte in ihrer Form und Größe stark eingeschränkt. Aus diesem Grund wurde die Webcam als optimale Lösung zur Farberkennung gewählt.

Am Ende des Förderbandes befinden sich die Sortierkästen, welche auf einem Fahrzeug montiert sind. Dieses kann mit Hilfe eines Motors bewegt werden. Wenn ein Objekt eine bestimmte Farbe hat, wird das Fahrzeug so gesteuert, dass die Sortierkästen in die richtige Position fahren und das Objekt in die entsprechende Kiste fällt. Um sicherzustellen, dass die Position der Sortierkästen immer korrekt ist, wird ein Ultraschallsensor verwendet, der kontinuierlich den Abstand zum Fahrzeug misst. Allerdings misst dieser auf kurze Distanz sehr ungenau, weshalb zusätzlich ein Tastsensor eingesetzt wird. Dieser Tastsensor dient als Anschlag und bewirkt bei Betätigung durch das Fahrzeug, dass dieses zum Stillstand kommt.

Schließlich werden alle Komponenten (Sensoren und Motoren) an den NXT-Stein angeschlossen, der als zentrale Steuereinheit fungiert.

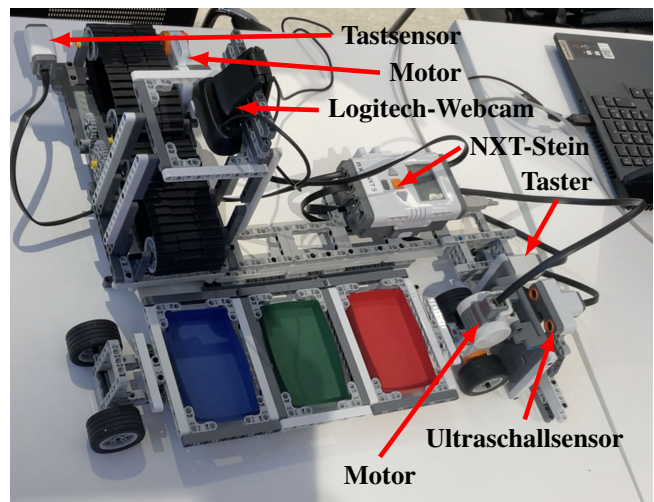


Abbildung 3. Aufbau der Farbsortiermaschine

C. Wie bekommt man aus einem Bild eine Farbe?

Gegenstand dieses Abschnittes ist die programmiertechnische Umsetzung, wobei eine Kamera zur Erkennung von Farben eingesetzt werden soll.

1) *Bild mit einer Kamera erstellen:* Der Beispielcode III-C1 zeigt die Erstellung eines Bildes mit den zugehörigen Farbwerten in MATLAB. Zunächst wird ein Kamera-Objekt mit dem Namen `camera` angelegt. In der nächsten Zeile des Codes wird mit der Kamera ein Bild erzeugt und mit dem `image`-Befehl in MATLAB angezeigt.

```
camera = videoinput('winvideo')
frame = getsnapshot(camera);
image(frame);
R = [frame(:, :, 1)];
G = [frame(:, :, 2)];
B = [frame(:, :, 3)];
```

Beispielcode: Zum Erstellen und Anzeigen eines Bildes sowie zum Auslesen von Farbwerten

2) *Farbwert aus einem Bild auslesen:* In MATLAB wird ein Koordinatensystem über das Bild gelegt, das es ermöglicht, für jedes Pixel den entsprechenden Farbwert mit Hilfe der `frame`-Funktion auszulesen. Die `frame`-Funktion besteht aus drei Komponenten (vgl. Abbildung 4), beginnend mit der X-Koordinate, gefolgt von der Y-Koordinate im Bild und schließlich dem Farbkanal (1 rot, 2 grün, 3 blau). Aufgrund der fehlenden Angabe konkreter X- und Y-Werte werden die Farbwerte des gesamten Bildes erfasst und in einer Matrix mit dem Namen `R` gespeichert.

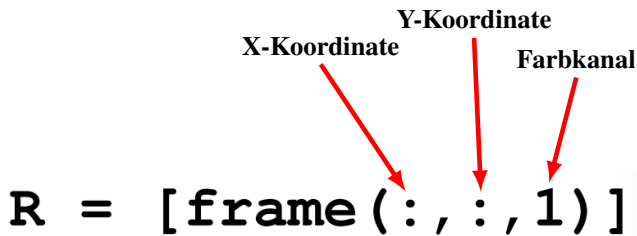


Abbildung 4. Aufbau der Frame Funktion

3) *Farbbereiche in einem Bild markieren:* Zunächst wird für jede vordefinierte Farbe (Rot-, Grün- und Blautöne) ein Farbbereich aus der Mischung der drei Grundfarbwerte (Rot-, Grün- und Blauwerte) erzeugt, um Farbbereiche in einem Bild zu kennzeichnen. Diese Werte werden anschließend in die `mask`-Funktion integriert (siehe die farbig hinterlegten Werte im Beispielcode III-C3). Dabei werden die Farbwerte aus dem gesamten Bild mit Hilfe der `frame`-Funktion ausgelesen. Liegen alle drei Farbwerte des Pixels im vordefinierten Bereich für Rot, Grün oder Blau, wird dieses Pixel durch die `mask`-Funktion markiert und erhält den Wert 1. Liegen die Werte außerhalb des Bereichs, erhält das Pixel den Wert Null. Alle Einsen und Nullen werden in einer neuen Matrix gespeichert.

```
mask = (frame(:, :, 1) >= 160)
      & (frame(:, :, 1) <= 255)
      & (frame(:, :, 2) <= 55) &
      (frame(:, :, 3) <= 55);
```

```
if any(mask(:))
    r = 1;
end
```

Beispielcode: Zur Markierung roter Farbbereiche mithilfe der `mask`-Funktion

Um zu prüfen, ob eine spezifische Farbe im Bild vorhanden ist, wird die `if`-Funktion im Beispielcode III-C3 angewendet. Dabei wird die neu erzeugte Matrix, welche ausschließlich aus Nullen und Einsen besteht, auf das Vorhandensein von Einsen untersucht. Das Vorkommen einer einzigen 1 in der Matrix ist bereits ausreichend, um die im Bild vorhandene Farbe zu erkennen.

D. Schnelle Verarbeitung von Bildern

Im Folgenden geht es darum, möglichst schnell Bilder zu generieren und diese jeweils auf die Anwesenheit einer der vordefinierten Farben zu überprüfen.

Der in Abbildung III-D dargestellte Code ermöglicht eine effiziente Erstellung und Verarbeitung der Daten. Mit Hilfe der `while`-Schleife können Bilder erzeugt werden, solange eine bestimmte Bedingung erfüllt ist. Allerdings kann mit dieser Methode nur etwa alle 3 Sekunden ein Bild erstellt werden.

Der Grund dafür ist, dass bei alleiniger Verwendung der Schleife bei jedem Durchlauf das Kameraobjekt geöffnet, eine Aufzeichnung gestartet, ein Bild erzeugt, die Aufzeichnung gestoppt und das Kameraobjekt wieder geschlossen werden müsste. Hinzu kommt die Datenverarbeitung, die sehr zeitintensiv ist.

Die Verwendung des Befehls `triggerconfig(camera, 'manual')` beschleunigt die Datenverarbeitung erheblich, da nur einmalig eine Verbindung zum Gerät aufgebaut und eine Konfiguration durchgeführt wird. Außerdem werden die Daten nur einmal gespeichert, was eine schnelle Verarbeitung in MATLAB ermöglicht. Durch die Verwendung des `trigger`-Befehls ist es möglich, jede Sekunde ein Bild zu erzeugen. Erst durch diese Methode ist eine schnelle Farberkennung mit einer Kamera realisierbar.

```
triggerconfig(camera, 'manual');
while (Bedingung)
    frame = getsnapshot(camera);
end
```

Beispielcode: Für ständige Bilderstellung

E. Herausforderungen

Im Zuge des Entwicklungsprozesses traten verschiedene Herausforderungen auf, die es zu bewältigen galt. Bei der Konstruktion der Farbsortiermaschine erwies sich zunächst die Konstruktion des Getriebes, einschließlich der Auswahl geeigneter Zahnräder, als anspruchsvoll. Auch die exakte Positionierung und Ausrichtung der Kamera war komplex, da dies die Belichtung des Objekts und die Größe des Kamerabildes beeinflusst. Die Stabilisierung und Fixierung des Ultraschall- und Tastsensors stellten sich ebenfalls als Schwierigkeit heraus, da sie notwendig waren, um die Sensoren auszurichten und ihre Stabilität gegenüber den Bewegungen

des Fahrzeugs zu gewährleisten.

Ebenso war die Interpretation der Messdaten des Ultraschallsensors herausfordernd, da dieser für die exakte Positionierung der Sortierkisten verantwortlich ist. Darüber hinaus erwies sich die Inkonsistenz der Messwerte des Ultraschallsensors auf kurze Distanz als Herausforderung, da für die Ausgangsposition eine kurze Distanz vom Fahrzeug zum Ultraschallsensor erforderlich war.

IV. ERGEBNISDISKUSSION

Am Ende waren die Herausforderungen gemeistert und eine funktionsfähige Farbsortiermaschine war das Ergebnis. Diese ist in der Lage, Objekte nach den Farben Rot, Blau und Grün mittels einer Kamera zu erkennen und in dafür vorgesehene Kisten zu sortieren.

Neben einfarbigen Objekten werden auch mehrfarbige Objekte erkannt und durch den Rückwärtslauf des Förderbandes zur Ausgangsposition zurückbefördert, wo sie dann manuell aussortiert werden können. Das Aussortieren von mehrfarbigen Objekten funktioniert allerdings nur im ersten Durchlauf, da die While-Schleife, die den Großteil des Programms umfasst, erst nach der Unterscheidung, ob mehrere Farben vorhanden sind, beginnt.

Ebenso besteht die Möglichkeit, die Bildverarbeitung zu beschleunigen, da diese fast nur noch durch die Prüfung, ob eine Farbe vorhanden ist, eingeschränkt wird. Dazu wäre es beispielsweise möglich, sich bei der Farberkennung auf die Bildmitte zu beschränken, wodurch deutlich weniger Pixel auf ihre Farbe hin überprüft werden müssten. Dies hätte zur Folge, dass Bilder in kürzeren Zeitabständen erstellt werden können und gleichzeitig vermieden wird, dass Objekte aufgrund zu großer Zeitintervalle bei der Bilderzeugung nicht erkannt werden.

Ebenfalls könnte durch die Integration weiterer Farben mit ihren jeweiligen Farbbereichen ausgeschlossen werden, dass bei mehrfarbigen Objekten nur eine der beiden Farben erkannt wird und diese somit in die falsche Kiste einsortiert werden. Dies war jedoch in diesem Fall nicht möglich, da die Länge des Kabels, das den Motor der Sortierkisten mit dem NXT-Stein verbindet, die Anzahl der Farben begrenzt. Zudem würde bei einer Erweiterung der Farbpalette die Bildverarbeitung wieder länger dauern.

Dies zeigt überaus deutlich, dass Prozesse oft ineinander greifen und sich somit gegenseitig beeinflussen, weshalb Verbesserungen immer im Gesamtsystem und nicht nur im Einzelschritt betrachtet werden müssen.

V. ZUSAMMENFASSUNG UND FAZIT

Zum Ende des LEGO-Praktikums war die Farbsortiermaschine, mit der Objekte nach ihrer Farbe sortiert werden können, fertiggestellt. Die Sortiermaschine hat noch viel Potenzial für Verbesserungen, wie bereits in der Ergebnisdiskussion festgestellt wurde. Die Umsetzung des selbst gesteckten Ziels wurde erreicht und das LEGO-Praktikum konnte erfolgreich abgeschlossen werden. Dabei wurde nicht nur die grundlegende Programmierung in MATLAB erlernt, sondern auch die Anwendung dieser Kenntnisse bei der Steuerung der selbstgebauten LEGO-Mindstorms-Roboter.

LITERATUR

- [1] MathWorks: *triggerconfig*, http://de.mathworks.com/help/imaq/imaqdevice.triggerconfig.html?s_tid=srchtitle_triggerconfig_1 (Stand: Februar 2023)
- [2] Mathworks: *Acquiring a Single Image in a Loop*, <https://de.mathworks.com/help/imaq/acquiring-a-single-image-in-a-loop.html> (Stand: Februar 2023)
- [3] Joachim Bone: NXT-Stein: <https://www.brick-shop.de/Elektrik-4951.html?language=de> (Stand: Februar 2023)
- [4] LEGO: Tastsensor: <https://www.lego.com/cdn/cs/set/assets/bltee56bc5628e9b8bf9843.jpg> (Stand: Februar 2023)
- [5] Amazone: LEGO NXT Ultraschallsensor Ultrasonic sensor: <https://www.amazon.de/LEGO-NXT-Ultraschallsensor-Ultrasonic-sensor/dp/B000PM8180> (Stand: Februar 2023)
- [6] Amazone: Logitech-C270-Webcam: <https://www.amazon.de/Logitech-C270-Webcam-720p-schwarz/dp/B01BGBJ8Y0> (Stand: Februar 2023)
- [7] Brick Owl: LEGO-Mindstorms EV3 Groß Motor: <https://www.brickowl.de/catalog/lego-%20ev3-%20large-%20servo-%20motor-%20C2%A0set-%20A045502> (Stand: Februar 2023)