



Politechnika
Śląska

PROJEKT INŻYNIERSKI

Robotyczny system sortowania klocków

Jakub PAJĄK

Nr albumu: 300566

Kierunek: Automatyka i Robotyka

Specjalność: Technologie Informacyjne w Automatyce i Robotyce

PROWADZĄCY PRACĘ

dr inż. Krzysztof Jaskot

KATEDRA Automatyki i Robotyki

Wydział Automatyki, Elektroniki i Informatyki

Gliwice 2024

Tytuł pracy

Robotyczny system sortowania klocków

Streszczenie

Celem projektu jest wykonanie autonomicznej platformy jezdnej, mającej na celu bezobsługowe sortowanie klocków spadających w określone miejsce z taśmociągu lub innego podajnika. Robot korzystając z możliwości wizji komputerowej będzie rozpoznawać kolor aktualnego klocka, a następnie na podstawie określonego koloru będzie wybierać trasę do odpowiedniego pojemnika. Domyślnie rozpoznawane będą trzy kolory (czerwony, zielony, niebieski) oraz trzy pojemniki, odpowiednie dla każdego koloru. Wymagania: znajomość programowania układów SBC (Raspberry Pi), Python, OpenCV

Słowa kluczowe

Robot mobilny, analiza wizyjna, Programowanie mikrokontrolerów

Thesis title

Robotic brick sorting system

Abstract

The goal of this project is to create an autonomous mobile platform designed for unattended sorting of blocks falling into a designated area from a conveyor belt or another feeder. The robot will use computer vision to recognize the color of the current block and, based on the identified color, will choose the path to the appropriate container. By default, three colors (red, green, blue) will be recognized, and there will be three containers, each corresponding to one of the colors. Requirements: knowledge of SBC programming (Raspberry Pi), Python, OpenCV

Key words

Mobile robot, visual analysis, microcontroller programming

Spis treści

1	Wstęp	1
1.1	Wprowadzenie w problem	1
1.2	Osadzenie problemu w dziedzinie	2
1.3	Cel pracy	2
1.4	Zakres pracy	2
1.5	Zwięzła charakterystyka rozdziałów	3
1.6	Wkład autora	3
2	Analiza tematu	5
2.1	Sformułowanie problemu	5
2.2	Osadzenie tematu w kontekście aktualnego stanu wiedzy (<i>state of the art</i>) .	5
2.3	Studia literaturowe	5
3	Wymagania i narzędzia	7
4	[Właściwy dla kierunku – np. Specyfikacja zewnętrzna]	9
5	[Właściwy dla kierunku – np. Specyfikacja wewnętrzna]	11
6	Weryfikacja i walidacja	13
7	Podsumowanie i wnioski	15
	Bibliografia	17
	Spis skrótów i symboli	21
	Źródła	23
	Lista dodatkowych plików, uzupełniających tekst pracy	25
	Spis rysunków	27
	Spis tabel	29

Rozdział 1

Wstęp

Niniejsza praca inżynierska koncentruje się na zaprojektowaniu i budowie autonomicznej platformy jezdnej, której zadaniem jest sortowanie klocków na podstawie ich koloru. W dobie rozwijającej się automatyzacji i robotyzacji coraz większą rolę odgrywają systemy zdolne do autonomicznego wykonywania zadań, które wcześniej wymagały udziału człowieka. Projekt ten wkomponowuje się w ten trend, mając na celu zautomatyzowanie procesu sortowania elementów w środowisku przemysłowym lub magazynowym. W tym rozdziale zostaną omówione kluczowe aspekty projektu, w tym wprowadzenie w problem, osadzenie go w odpowiednim kontekście naukowym oraz zakres i cele pracy.

1.1 Wprowadzenie w problem

Problem, który stara się rozwiązać niniejszy projekt, dotyczy automatyzacji procesów sortowania w różnych środowiskach przemysłowych, takich jak linie produkcyjne czy magazyny. W tradycyjnym podejściu sortowanie odbywa się ręcznie lub z wykorzystaniem prostych mechanicznych sorterów, które często są mało elastyczne i kosztowne w utrzymaniu. W odpowiedzi na to wyzwanie, projekt ten przewiduje stworzenie autonomicznego robota mobilnego, wykorzystującego kamerę i algorytmy wizji komputerowej do rozpoznawania kolorów klocków oraz następnie do przewiezienia ich do odpowiednich pojemników.

Takie rozwiązanie jest szczególnie przydatne w sytuacjach, gdzie istnieje potrzeba dynamicznej zmiany parametrów sortowania lub tam, gdzie wymagana jest szybka reakcja na zmiany w procesie produkcyjnym. Automatyzacja pozwala również na zmniejszenie kosztów operacyjnych, zwiększenie efektywności oraz redukcję błędów ludzkich w procesie sortowania.

1.2 Osadzenie problemu w dziedzinie

Tematyka projektu wpisuje się w kilka dynamicznie rozwijających się dziedzin, takich jak robotyka mobilna, automatyzacja przemysłowa oraz wizja komputerowa. Roboty mobilne, zdolne do samodzielnej nawigacji oraz wykonywania skomplikowanych zadań, znajdują coraz szersze zastosowanie w przemyśle, logistyce, a także w gospodarstwach domowych. W kontekście niniejszego projektu, kluczową rolę odgrywa zastosowanie odometrii oraz systemów napędowych, które pozwalają na precyzyjną kontrolę ruchu robota.

Wizja komputerowa, z kolei, umożliwia rozpoznawanie obiektów na podstawie ich cech wizualnych, takich jak kolor czy kształt. Technologia ta, oparta na bibliotekach takich jak OpenCV, stała się dostępna nawet dla małych systemów wbudowanych, takich jak Raspberry Pi. W niniejszym projekcie robot wykorzystuje kamerę do rejestrowania obrazów klocków, które następnie są analizowane w czasie rzeczywistym w celu określenia koloru, a na tej podstawie podejmowana jest decyzja gdzie analizowany klocek powinien zostać przetransportowany.

1.3 Cel pracy

Głównym celem projektu jest zaprojektowanie i zbudowanie autonomicznej platformy jezdnej, która potrafi sortować klocki na podstawie ich koloru. Robot będzie wykorzystywał kamerę oraz algorytmy wizji komputerowej do identyfikacji kolorów takich jak czerwony, zielony oraz niebieski, a następnie będzie odpowiednio przekierowywał klocki do odpowiednich pojemników. Proces ten ma odbywać się w pełni automatycznie, bez potrzeby ingerencji człowieka.

Dodatkowym celem jest stworzenie systemu kontroli, który pozwoli na bieżąco monitorować pracę robota oraz ewentualnie wprowadzać modyfikacje w jego działaniu. System powinien być elastyczny, umożliwiając łatwe rozszerzenie o dodatkowe funkcje, takie jak rozpoznawanie większej liczby kolorów lub innych cech klocków.

1.4 Zakres pracy

Projekt obejmuje szeroki zakres działań, zarówno w zakresie sprzętowym, jak i programistycznym. Poniżej przedstawiono główne etapy realizacji:

- Zbudowanie fizycznej platformy jezdnej, w tym konstrukcji mechanicznej oraz systemu napędowego z enkoderami,
- Implementacja systemu rozpoznawania kolorów za pomocą kamery oraz algorytmów wizji komputerowej z użyciem Raspberry Pi i biblioteki OpenCV,

- Opracowanie algorytmu podejmowania decyzji na podstawie rozpoznanego koloru i przekierowania klocka do odpowiedniego pojemnika,
- Testowanie i optymalizacja działania systemu, aby zapewnić jego poprawne funkcjonowanie w rzeczywistych warunkach, takich jak linie produkcyjne czy magazyny.

Zakres projektu przewiduje także możliwość dalszej rozbudowy o nowe funkcjonalności, co zwiększa elastyczność i potencjalne zastosowanie platformy w innych dziedzinach, takich jak automatyczne sortowanie elementów na podstawie kształtu czy materiału.

1.5 Zwięzła charakterystyka rozdziałów

Struktura pracy została podzielona na następujące rozdziały:

- **Rozdział 1: Wstęp** – Przedstawienie problemu, celów oraz zakresu projektu.
- **Rozdział 2: Analiza tematu** – Przegląd dostępnych rozwiązań, technik oraz technologii stosowanych w podobnych systemach, a także omówienie aktualnego stanu wiedzy w tej dziedzinie.
- **Rozdział 3: Wymagania i narzędzia** – Opis wymagań projektowych oraz narzędzi, takich jak Raspberry Pi, OpenCV oraz Python, które zostaną użyte do realizacji projektu.
- **Rozdział 4: Specyfikacja zewnętrzna** – Opis funkcjonalności systemu z perspektywy użytkownika, w tym interfejsu oraz interakcji z otoczeniem.
- **Rozdział 5: Specyfikacja wewnętrzna** – Szczegółowa analiza architektury wewnętrznej systemu, w tym struktury oprogramowania i integracji z komponentami sprzętowymi.
- **Rozdział 6: Weryfikacja i walidacja** – Opis przeprowadzonych testów, ocena efektywności systemu oraz zgodności z założeniami projektowymi.
- **Rozdział 7: Podsumowanie i wnioski** – Zakończenie projektu, omówienie osiągniętych celów, a także wskazanie możliwych kierunków rozwoju systemu.

1.6 Wkład autora

Projekt został w pełni zrealizowany przez autora. Autor odpowiada za wszystkie etapy pracy, począwszy od projektowania i budowy platformy, poprzez implementację algorytmów rozpoznawania kolorów i sterowania ruchem robota, aż po testowanie i optymalizację systemu. Ponadto, autor zaprojektował system kontrolny, który pozwala na monitorowanie pracy robota w czasie rzeczywistym oraz umożliwia wprowadzenie ewentualnych korekt.

Rozdział 2

Analiza tematu

W rozdziale tym zostanie przeanalizowany temat projektu, z uwzględnieniem aktualnego stanu wiedzy oraz znanych rozwiązań dotyczących automatycznych systemów sortowania za pomocą wizji komputerowej.

2.1 Sformułowanie problemu

Głównym problemem rozwiązywanym w tym projekcie jest automatyzacja procesu sortowania obiektów według ich cech wizualnych, w tym przypadku koloru. Tradycyjne metody sortowania wymagają interwencji człowieka lub stosowania mechanicznych sorterów, które często są mniej elastyczne i mniej precyzyjne w identyfikacji złożonych cech. Celem projektu jest wykorzystanie technologii wizji komputerowej, aby umożliwić robotowi autonomiczne rozpoznawanie i sortowanie klocków według koloru.

2.2 Osadzenie tematu w kontekście aktualnego stanu wiedzy (*state of the art*)

Temat projektu osadza się w dziedzinach robotyki mobilnej oraz wizji komputerowej, które rozwijają się bardzo dynamicznie w ostatnich latach. Rozwiązania oparte na wizji komputerowej, zwłaszcza w połączeniu z platformami SBC (Single-Board Computer), takimi jak Raspberry Pi, stają się coraz bardziej popularne w automatyzacji procesów przemysłowych i logistycznych. Wykorzystanie takich narzędzi jak OpenCV w projektach robotycznych pozwala na zwiększenie elastyczności i precyzji systemów sortujących.

2.3 Studia literaturowe

Studia literaturowe będą obejmowały analizę dostępnych rozwiązań naukowych i technicznych dotyczących autonomicznych systemów sortowania oraz technologii wizji kompu-

terowej. Przegląd literatury pozwoli zidentyfikować kluczowe algorytmy wykorzystywane do rozpoznawania kolorów oraz technologie stosowane w systemach robotycznych tego typu. Ważne będzie także porównanie różnych podejść do implementacji i optymalizacji systemów autonomicznych sorterów.

[3, 4, 8, 7, 1, 2, 5, 6]

Rozdział 3

Wymagania i narzędzia

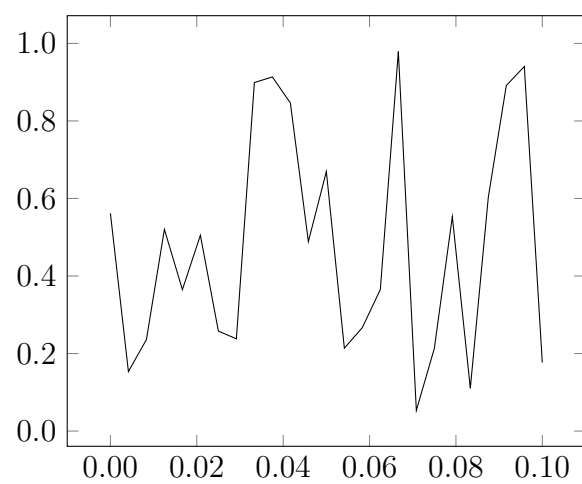
- wymagania funkcjonalne i нефункционалне
- przypadki użycia (diagramy UML) – dla prac, w których mają zastosowanie
- opis narzędzi, metod eksperymentalnych, metod modelowania itp.
- metodyka pracy nad projektowaniem i implementacją – dla prac, w których ma to zastosowanie

Rozdział 4

[Właściwy dla kierunku – np. Specyfikacja zewnętrzna]

Jeśli „Specyfikacja zewnętrzna”:

- wymagania sprzętowe i programowe
- sposób instalacji
- sposób aktywacji
- kategorie użytkowników
- sposób obsługi
- administracja systemem
- kwestie bezpieczeństwa
- przykład działania
- scenariusze korzystania z systemu (ilustrowane zrzutami z ekranu lub generowanymi dokumentami)



Rysunek 4.1: Podpis rysunku po rysunkiem.

Rozdział 5

[Właściwy dla kierunku – np. Specyfikacja wewnętrzna]

Jeśli „Specyfikacja wewnętrzna”:

- przedstawienie idei
- architektura systemu
- opis struktur danych (i organizacji baz danych)
- komponenty, moduły, biblioteki, przegląd ważniejszych klas (jeśli występują)
- przegląd ważniejszych algorytmów (jeśli występują)
- szczegóły implementacji wybranych fragmentów, zastosowane wzorce projektowe
- diagramy UML

Krótką wstawka kodu w linii tekstu jest możliwa, np. **int a;** (biblioteka `listings`). Dłuższe fragmenty lepiej jest umieszczać jako rysunek, np. kod na rys 5.1, a naprawdę długie fragmenty – w załączniku.

```
1 class test : public basic
2 {
3     public:
4         test (int a);
5         friend std::ostream operator<<(std::ostream & s,
6                                         const test & t);
7     protected:
8         int _a;
9
10 };
```

Rysunek 5.1: Pseudokod w `listings`.

Rozdział 6

Weryfikacja i walidacja

- sposób testowania w ramach pracy (np. odniesienie do modelu V)
- organizacja eksperymentów
- przypadki testowe zakres testowania (pełny/niepełny)
- wykryte i usunięte błędy
- opcjonalnie wyniki badań eksperymentalnych

Tabela 6.1: Nagłówek tabeli jest nad tabelą.

ζ	metoda						
	alg. 1	alg. 2	alg. 3			alg. 4, $\gamma = 2$	
			$\alpha = 1.5$	$\alpha = 2$	$\alpha = 3$	$\beta = 0.1$	$\beta = -0.1$
0	8.3250	1.45305	7.5791	14.8517	20.0028	1.16396	1.1365
5	0.6111	2.27126	6.9952	13.8560	18.6064	1.18659	1.1630
10	11.6126	2.69218	6.2520	12.5202	16.8278	1.23180	1.2045
15	0.5665	2.95046	5.7753	11.4588	15.4837	1.25131	1.2614
20	15.8728	3.07225	5.3071	10.3935	13.8738	1.25307	1.2217
25	0.9791	3.19034	5.4575	9.9533	13.0721	1.27104	1.2640
30	2.0228	3.27474	5.7461	9.7164	12.2637	1.33404	1.3209
35	13.4210	3.36086	6.6735	10.0442	12.0270	1.35385	1.3059
40	13.2226	3.36420	7.7248	10.4495	12.0379	1.34919	1.2768
45	12.8445	3.47436	8.5539	10.8552	12.2773	1.42303	1.4362
50	12.9245	3.58228	9.2702	11.2183	12.3990	1.40922	1.3724

Rozdział 7

Podsumowanie i wnioski

- uzyskane wyniki w świetle postawionych celów i zdefiniowanych wyżej wymagań
- kierunki ewentualnych danych prac (rozbudowa funkcjonalna ...)
- problemy napotkane w trakcie pracy

Bibliografia

- [1] Aditi Bajaj i Jyotsna Sharma. „Computer Vision for Color Detection”. W: *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)* (2020), s. 53–59.
- [2] Gary Bradski i Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. United States of America, Sebastopol: O'Reilly Media, 2008. ISBN: 978-0-596-51613-0.
- [3] B. Cottenceau. *Introduction to Arduino*.
- [4] Raspberry Pi Ltd. *The Picamera2 Library. A libcamera-based Python library for Raspberry Pi cameras*.
- [5] Octavian Bologa Mihai Crenganis. „PID CONTROLLER FOR A DIFFERENTIAL STEERING MOBILE PLATFORM”. W: *Konferencji ASTR*. 2015, s. 6.
- [6] Imię Nazwisko i Imię Nazwisko. *Tytuł strony internetowej*. 2021. URL: <http://gdzies/w/internecie/internet.html> (term. wiz. 30.09.2021).
- [7] SHRIPAD G. DESAI PRINCE KUMAR. „Interrupts in The Microcontroller”. W: *IRE Journals* 4.4 (2021), s. 166–169.
- [8] Djemel Ziou i Salvatore Tabbone. „Edge Detection Techniques-An Overvie”. W: *£#1085 / Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications* 8.4 (1998), s. 537–559.

Dodatki

Spis skrótów i symboli

DNA kwas deoksyrybonukleinowy (ang. *deoxyribonucleic acid*)

MVC model – widok – kontroler (ang. *model-view-controller*)

N liczebność zbioru danych

μ stopnień przyleżności do zbioru

\mathbb{E} zbiór krawędzi grafu

\mathcal{L} transformata Laplace’a

Źródła

Jeżeli w pracy konieczne jest umieszczenie długich fragmentów kodu źródłowego, należy je przenieść w to miejsce.

```
1 if (_nClusters < 1)
2     throw std::string ("unknown_number_of_clusters");
3 if (_nIterations < 1 and _epsilon < 0)
4     throw std::string ("You should set a maximal number of
        iteration or minimal difference — epsilon.");
5 if (_nIterations > 0 and _epsilon > 0)
6     throw std::string ("Both number of iterations and minimal
        epsilon set — you should set either number of iterations
        or minimal epsilon.");
```

Lista dodatkowych plików, uzupełniających tekst pracy

W systemie do pracy dołączono dodatkowe pliki zawierające:

- źródła programu,
- dane testowe,
- film pokazujący działanie opracowanego oprogramowania lub zaprojektowanego i wykonanego urządzenia,
- itp.

Spis rysunków

4.1	Podpis rysunku po rysunkiem.	10
5.1	Pseudokod w <code>listings</code>	12

Spis tabel

6.1	Nagłówek tabeli jest nad tabelą.	14
-----	--	----