

Software-Migration für die robotergestützte Lagerverwaltung mit kameragestützten Validierungsprozessen

Erstellung einer integrierten Python-Anwendung mit kameragestützten
Validierungsprozessen in der Industrie 4.0-Plattform Modellfabrik μ Plant

Lennart Schink

May 30, 2023

Contents

1	Einleitung	3
1.1	Einführung und Motivation	3
1.2	Zielsetzung	3
2	Theoretischer Hintergrund	4
2.1	Die Programmiersprache Python	4
2.2	GUI Programmierung mit PySide6 und Qt QuickQML	4
2.3	OPC UA	4
2.4	TCP/IP mittels websocket Bibliothek	4
2.5	RFID	4
3	Stand der Forschung	5
3.1	Merkmalerkennung	5
3.2	arUco Marker	5
3.3	Neuronale Netze ?	5
4	Methodik	6
4.1	Architektur der Python-Anwendung	6
4.2	Implementierung der Lagerverwaltung	6
4.3	Implementierung der Merkmalerkennung	6
4.3.1	Bildgewinnung	6
4.3.2	Algorithmus zur Merkmalerkennung	6
4.4	Maßnahmen zur Fehlerbehandlung	6
5	Ergebnisse	7

6 Diskussion	8
7 Fazit	9
8 Ausblick	10
9 Danksagung	11
10 Anhang	12

1 Einleitung

1.1 Einführung und Motivation

1.2 Zielsetzung

Erstellen einer Software die drei bestehende Programme integriert. Die ganze Software wird nach Python emigriert. Es wird eine Software implementiert, die

- zwei Kameras verwaltet
 - eine hochauflösende: Erfasst den gesamten Raum, Bild wird anhand markern aufgeteilt, und versucht in den Segmenten die marker der Becher zu erkennen
 - eine weitere Kamera wird auf dem Arm montiert. Bild dienst zur lokalen Erkennung der Marker vor/ im Greifer.
- Auf Aufforderung des Benutzers oder in festgelegten Abständen oder aufgrund der Türöffnung eine automatisierte Inventur durchführt:
 - Versuch die Inventur über die hochauflösende Kamera durchzuführen.
 - Zur Verifizierung / bei Nichterkennung oder auf Aufforderung:
- -Fahrt des Greifers vor dem Regal in einer Haltung die das bestmögliche Erkennen der Becher ermöglicht
- -Paletten werden auf die Werkbank gestellt und dort Marker auf den Bechern erkannt.

2 Theoretischer Hintergrund

2.1 Die Programmiersprache Python

-kurzes Vorstellen der Sprache -verwendete Bibliotheken außer nachfolgende untertitel

2.2 GUI Programmierung mit PySide6 und Qt QuickQML

- Datenmodelle - Signal/Slot - Prinzip

2.3 OPC UA

- Konzept - Code

2.4 TCP/IP mittels websocket Bibliothek

- Konzept - Code

2.5 RFID

- Konzept - Code - Erkennungsraten?

3 Stand der Forschung

3.1 Merkmalerkennung

3.2 arUco Marker

3.3 Neuronale Netze ?

4 Methodik

4.1 Architektur der Python-Anwendung

- Bezugnahme auf Semesterarbeit
- Vorstellen der neuen Softwarearchitektur

4.2 Implementierung der Lagerverwaltung

- Besonderheiten in der Implementierung
- Controller \rightarrow DatenModell \rightarrow GUI-Element Zuordnung? oder Klassen/Objektdiagramm?

4.3 Implementierung der Merkmalerkennung

4.3.1 Bildgewinnung

- Datengenerierung von den Bildsensoren
- Preprozessing und Bereitstellung an GUI

4.3.2 Algorithmus zur Merkmalerkennung

- openCV cv2 arUco Generierung
- openCV cv2 arUco Erkennung
- angewandte Bildverarbeitungsmethoden:
 - - Bildaufteilung
 - - verwendete Filter: Gauß, Dilatation, Custom-Filter

4.4 Maßnahmen zur Fehlerbehandlung

- Becher nicht erkannt
- Becher fälschlicherweise erkannt
- nicht alle Becher in der globalen Erkennung erkannt
- mehr als 36 Becher in der globalen Erkennung erkannt

5 Ergebnisse

6 Diskussion

7 Fazit

8 Ausblick

9 Danksagung

10 Anhang