

# Entwicklung eines flexiblen, inkrementell lernenden Programmiersystems für kollaborative Roboterapplikationen

Eine Kombination von kollaborativer Robotik, Machine-Learning und -Vision für industriell relevante Demonstrationen

Andrea Giusti<sup>1</sup>, Dieter Steiner<sup>1</sup>, Sebastian Bertoli<sup>1</sup>, Dominik T. Matt<sup>1,2</sup>

<sup>1</sup>Fraunhofer Italia Research s.c.a.r.l., Bozen; <sup>2</sup>Industrial Engineering and Automation, Freie Universität Bozen

## 1. Einleitung und State of the Art

Erfahrungen aus der Projektakquise und laufenden Forschungsprojekten zeigen, dass für viele Kunden, insbesondere aus dem Bereich der kleinen und mittleren Unternehmen (KMU), der effektivste Weg zum schnellen Verständnis die Praxis ist. Was Stunden von Gesprächen und Präsentation nicht erklären – das können fünf Minuten einer Live-Demonstration (Matt et al. 2018). Dies zeigt, wie wichtig es ist, Plattformen zu realisieren, auf denen komplexe Konzepte aus der Grundlagenforschung in einfach verständlichen Paketen praktisch erfahrbar sind.

Betrachtet man beispielsweise die potenziellen Vorteile der Automatisierung für KMUs (z.B. verbesserte Time-to-Market, Effizienz und Qualität), kann ein Demonstrator ein überzeugendes Mittel sein, um den potenziellen Wandel von klassischen starren Automationssystemen hin zu modernen, flexibleren Gegenstücken aufzuzeigen. In diesem Zusammenhang können KMUs direkt von einem Demonstrator profitieren, indem sie mit flexiblen Produktionssystemen vertraut gemacht werden. Sie können dadurch ihr Investitionsrisiko in die Automation verringern und ihre Anpassungsfähigkeit an Marktveränderungen erhöhen. Im Vergleich zu herkömmlichen Robotersystemen erlauben moderne, flexible Produktionsanlagen die Anpassung an neue Aufgaben ohne großen Aufwand, Kosten oder Standzeiten – dadurch steigt ihr Nutzen und Return of Investment (ROI).

Im Laufe der letzten Jahre beobachten wir ein zunehmendes Interesse an der Weiterentwicklung klassischer Produktionssysteme unter Einbeziehung moderner Konzepte wie Digitalisierung, kollaborativer Robotik und künstlicher Intelligenz (KI), welche als Eckpfeiler der Vierten industriellen Revolution angesehen werden, siehe z.B. (Mittal et al. 2016). Diese Entwicklung zielt auf mehr Flexibilität, Effizienz und verbesserte Arbeitsbedingungen und wird auch durch verschiedene Investitions- und Gründungsprogramme weltweit gefördert (siehe z.B. "Piano Nazionale Industria 4.0" in Italien, "Industrie 4.0" in Deutschland und "Smart Manufacturing" in den USA).

Zu den wegbereitenden Arbeiten, welche die Verwendung intelligenter Systeme in der Fertigung begründen, gehört (Hatvany/Nemes 1978), in dem das Konzept eines Intelligent Manufacturing Systems (IMS) eingeführt wurde. In dieser Arbeit wird die wachsende Komplexität von Produktionssystemen bis hin zur flexiblen Automation vorausgesehen. Damals waren jedoch die verfügbaren Rechnersysteme noch nicht in der Lage, die großen Mengen an Informationen zu verarbeiten welche notwendig sind, um ausreichend komplexe Modelle zu erstellen und sie effektiv zu nutzen (Sharp et. al. 2018). In den letzten zwei Jahrzehnten stieg jedoch die verfügbare Rechenleistung in Embedded Systems und neue Methoden und Hilfsmittel (wie das Industrial Internet of Things (Xu et. al 2014) und Cloud-Computing (Tao et. al 2014)) wurden verfügbar, welche nun die Umsetzung von IMSs in der modernen Industrie begünstigen.

Es gibt vielerlei Methoden der KI welche zur erfolgreichen Lösung realer Probleme in der Produktion eingesetzt werden können. Eine vollständige Beschreibung dieser Methoden findet sich in (Russel/Norvig 2009). Die grundlegenden Konzepte mit hohem Anwendungspotenzial lassen sich in drei Kategorien einteilen: (i) Methoden, die darauf abzielen Probleme zu lösen welche auf einer vorbestimmten Wissensbasis beruhen, (ii) Methoden die zusätzlich den Unsicherheitsfaktor von realen Daten und Sensorsystemen berücksichtigen und (iii) Methoden, für die beide vorherigen Kategorien fundamental sind und darauf abzielen, automatisch das notwendige Wissen zu erwerben, damit das intelligente System seine Aufgabe erfüllen und sich an unvorhergesehene Szenarien anpassen kann. Die dritte Kategorie wird im Folgenden zusammenfassend als *Machine Learning* (ML) bezeichnet und gewinnt in der Praxis zunehmend an Bedeutung.

Die Vorteile und Herausforderungen für den Einsatz von ML-Methoden in der Fertigung sind in (Wuest 2016) ausführlich beschrieben. Zusammenfassend bieten ML-Techniken die Möglichkeit, hochdimensionale Daten zu verarbeiten, implizite Zusammenhänge und Muster in großen Datenmengen zu entdecken (z.B. um Vorhersagen zu treffen) und die Möglichkeit, eine kontinuierliche Qualitätsverbesserung in komplexen Szenarien durchzuführen. Die Herausforderungen für ihre Anwendungen werden von der oft begrenzten Verfügbarkeit, Qualität und Vollständigkeit der Datensätze dominiert, was sich negativ auf die erzielbaren Ergebnisse auswirken kann. Eine große Auswirkung auf die Qualität der Ergebnisse hat auch die Datenvorverarbeitung, da ML-Algorithmen hier sehr sensibel reagieren können. Weitere gemeinsame Herausforderungen sind die Auswahl der am besten geeigneten ML-Algorithmen für bestimmte Probleme und die korrekte Interpretation von Ergebnissen, welche z.B. durch Overfitting und Bias beeinflusst werden können.

Viele Anwendungen von KI und insbesondere von ML-Techniken in der Fertigung wurden in umfangreichen Studien wie (Monostori 2003) und (Pham/Afifi 2005) vorgestellt und gesammelt. Neuere Entwicklungen von ML-Anwendungen in der Produktion wurden in (Wuest et al. 2016) und (Sharp et al. 2018) diskutiert. Eine besonders interessante Anwendung für maschinelles Lernen in der Robotik ist das *Imitation Learning*, welches darauf abzielt, Roboter ausschließlich anhand von Demonstrationen zu programmieren (Billard et al. 2016). Zu den neuesten Entwicklungen in diesem Bereich gehört die Arbeit in (Giusti et al. 2018). Dabei wird Imitation Learning mit modularer Robotik kombiniert, um die Anwendbarkeit für industriell relevante Aufgaben aufzuzeigen und eine grundlegende Basis für ein neues Framework zur flexiblen Fertigung zu bilden.

Der Absatz von Industrierobotern wächst kontinuierlich, ihre Anwendung beschränkt sich nach wie vor überwiegend auf die Automobil- und Elektronikindustrie (Zanchettin et al. 2018). Dies kann damit zusammenhängen, dass das Risiko von Investitionen in Automatisierungssysteme mit kostspieliger oder gar unpraktischer Anpassung an Produktionsänderungen nach wie vor ein Hindernis für KMUs darstellt. Ein vielversprechendes Mittel zur Risikoreduzierung in diesem Zusammenhang ist die Erhöhung der Flexibilität von Roboterlösungen, die KMUs einen einzigartigen Vorteil bei der Massenanpassung bringt. Ein moderner Ansatz zur Steigerung der Flexibilität von Produktionssystemen ist die Realisierung industrieller Umgebungen, in welchen Mensch und Roboter kooperieren können, während sie sich den selben Arbeitsbereich tei-

len (Robla-Gomez et al. 2017). Dieser Ansatz ermöglicht die Kombination von Kraft, Wiederholbarkeit, Präzision und Ausdauer von Robotern mit den kognitiven Fähigkeiten und der Fingerfertigkeit des Menschen. Eine solche Kombination war bisher in der klassischen Industrierobotik nur eingeschränkt möglich und erforderte aus Sicherheitsgründen eine räumliche Trennung von Mensch und Maschine. In den letzten Jahrzehnten wurden viele Forschungsergebnisse im Bereich der physischen Mensch-Roboter-Interaktion (oder kollaborative Roboter) vorgestellt (siehe z.B. (Haddadin/Croft 2016)). Leichte und konforme Roboter wurden ebenso zur Verfügung gestellt wie Steuerungssysteme, welche die Sicherheit von Menschen in unmittelbarer Nähe von Robotern gewährleisten, während sie sich den selben Arbeitsbereich teilen (siehe z.B. (Zanchettin et al. 2016) und (Pereira/Althoff 2017)).

Im Gegensatz zur klassischen Industrierobotik stellen gemeinsame Arbeitsbereiche von Mensch und Robotern neue Sicherheitsanforderungen. Diese neuen Anforderungen sind in den Normen ISO 10218-1, ISO 10218-2 und der neueren technischen Spezifikation ISO/TS 15066 zusammengefasst. Neben der Sicherheit wird von kollaborativen Robotern auch eine benutzerfreundliche Programmierung für häufig wechselnde Aufgaben erwartet. Letzteres kann durch die Einbeziehung von maschinellen Lerntechniken wie dem bereits erwähnten Imitation Learning erreicht werden.

Fehlende Erfahrungen in KMUs mit den oben genannten Ansätzen können zu einer weitreichenden Einschränkung der Umsetzbarkeit führen. Um einen Beitrag zur Beseitigung dieser Herausforderung zu leisten, entwickeln wir einen Demonstrator, welcher praktische Anwendungen in der Fertigung für kollaborative Robotik und maschinelle Lerntechniken zeigt. Unser Demonstrator soll zeigen, wie ein Mensch und ein Robotersystem bei industriell relevanten Aufgaben zusammenarbeiten können, wobei der Roboter laufend vom Menschen lernt, wie er ihn besser unterstützen kann. Dies ermöglicht im Laufe der Zeit eine Reduzierung des menschlichen Arbeitsaufwands, vorausgesetzt, dass effektive Systeme für Machine-Vision und Incremental-Learning eingesetzt werden. Der Demonstrator soll zeigen, wie ein menschlicher Mitarbeiter und ein Roboter zusammenarbeiten und einander helfen können - der Roboter als unermüdlicher Arbeiter, der sich um die monotone Aufgabe des Sortierens von farbigen Würfeln kümmert, und der menschliche Mitarbeiter, welcher die Entscheidungen für das Unterrichten trifft und das Führen des Roboters übernimmt, wenn das Lernsystem nicht in der Lage ist eine sichere Entscheidung zu treffen.

## 2. Konzept

In diesem Abschnitt erläutern wir das Grundkonzept des zuvor vorgestellten Demonstrators, seine Funktionen und die Art und Weise, wie sie umgesetzt werden, unter Berücksichtigung sowohl der Hardware- als auch der Softwareseite.

### 2.1. Idee des Demonstrators

Auch wenn heutzutage eine hohe Flexibilität von robotischen Anwendungen erreicht werden kann, bleibt die Programmierung komplexer Aufgaben sehr zeitaufwändig. Dies wird umso schwieriger, wenn man unstrukturierte Szenarien betrachtet, in denen Menschen und Roboter physisch interagieren oder zusammenarbeiten können. Dies liegt vor allem an den möglichen Variationen der geometrischen Parameter der Aufgabe, die typischerweise bei der Programmierung des Robotersystems für klassische Anwendungen als bekannt und festgelegt angenommen werden und für neuartige Fälle eine Umprogrammierung erfordern. Eine Möglichkeit, diese Einschränkungen zu umgehen bietet das als *Imitation Learning* oder Programmieren durch Demonstration genannte Forschungsgebiet.

Das z.B. in (Billard et al. 2016) betrachtete Framework zielt auf einen vollständigen Paradigmenwechsel von der klassischen Programmierung von Robotern (z.B. mit Hilfe einer Programmiersprache oder eines Handterminals) hin zu Roboterprogrammierung, welche nur auf Demonstration beruht. Abgestimmt auf diesen Forschungsbereich zielen wir auf die Realisierung eines Systems, das nicht völlig unprogrammiert ist, aber flexible Operationen für eine Klasse von zuvor definierten Anwendungen ermöglicht, die sich auf Demonstrationen und die Zusammenarbeit mit einem Menschen stützen. Die hier betrachtete Anwendung ist die Sortierung von Bauteilen, die auf einem Förderband transportiert werden, in verschiedene Kisten. Wir bleiben hier sehr flexibel, da wir Variationen in Farbe, Form, Pose und Zuständen der zu sortierenden Objekte zulassen. Die industrielle Relevanz des gewählten Sortier-Demonstrators wird sofort deutlich, wenn man sich die Aufgabe ansieht, nach der (automatischen) maschinellen Bearbeitung oder dem Laserschneiden verschiedene Bauteile für unterschiedliche Aufträge zu sortieren oder fehlerhafte, bzw. nicht den Qualitätsansprüchen genügende Bauteile von einem Förderband auszusortieren. Insbesondere erlaubt uns die betrachtete Roboterplattform mit einfachen Testfällen (z.B. zunächst nur mit Farbvariabilität) zu beginnen, aber später auch komplexere Aufgaben zu bearbeiten (z.B. auch mit Formen zusätzlich zu Farben), welche robustere Verallgemeinerungsfähigkeiten erfordern.

Die Grundfunktionen des vorgeschlagenen Demonstrators können wie folgt beschrieben werden. Zu Beginn der Arbeitsphase weiß das Robotersystem nur, dass eine Pick-and-Place-Aufgabe von Objekten vom Transportband zu vorgegebenen Boxen durchgeführt werden soll. Zusätzlich wurde das Robotersystem unter Berücksichtigung der Sicherheitsvorschriften (ISO/TS 15066 2016) vorprogrammiert (z.B. sicheres Unterbrechen der Bewegung, wenn ein Mitarbeiter, der den Arbeitsbereich mitbenutzt, gefährdet ist). In der oben genannten technischen Spezifikation werden vier Methoden des kollaborativen Betriebs berücksichtigt: Handführung, sicherheitsgerichteter, überwachter Stopp, Geschwindigkeits- und Abstandsüberwachung sowie Leistungs- und Kraftbegrenzung.

Bei der Handführungsmethode kann der Benutzer den Roboter direkt über Betätigungseinrichtungen am (oder in der Nähe des) Endeffektors des Roboters bewegen. Damit lassen sich einfache Aufgaben ohne autonome Anpassungsfähigkeit in einem Record und Replay-Modus demonstrieren. Bei der sicherheitsgerichteten überwachten Stoppmethode muss die Bewegung des Roboters im kollaborativen Arbeitsbereich gestoppt werden, wenn ein Mensch den kollaborativen Arbeitsbereich betritt. Die ursprüngliche Bewegung kann erst wieder aufgenommen werden, nachdem der Mensch diesen Arbeitsbereich wieder verlassen hat. Dieser Umstand impliziert, dass der Beitrag des Roboters zu der gemeinsamen Arbeit erheblich eingeschränkt wird, wenn sich sowohl der Roboter als auch der Anwender häufig im gemeinsamen Arbeitsbereich befinden. Die Geschwindigkeits- und Abstandsmethode hingegen ermöglicht gleichzeitige Präsenz des Roboters und des Anwenders im gemeinsamen Arbeitsbereich. Bewegungen des Robotersystems können ausgeführt werden solange ein genügender Sicherheitsabstand zwischen dem Robotersystem und dem Anwender eingehalten wird. Im Gegensatz zum sicherheitsgerichteten überwachten Stopp, bei dem nur der menschliche Standort überwacht wird, ist die Geschwindigkeits- und Abstandsmethode schwieriger zu realisieren, da sie zusätzlich die Position der Roboterglieder (und eventuell auch deren Geschwindigkeit) berücksichtigt. Diese Methode verbessert jedoch den potenziellen Beitrag des Robotersystems zu den gemeinsamen Arbeiten und reduziert die Anzahl der notwendigen Roboterstopps. Die Leistungs- und Kraftbegrenzungsmethode erlaubt sogar den physischen Kontakt zwischen dem Robotersystem und dem Anwender während einer Bewegung. Diese Methode kann dann eingesetzt werden, wenn aufgabenspezifische Risikobewertungs- und -reduzierungsverfahren durchgeführt wurden und das Robotersystem so ausgelegt ist, dass es diesen Betriebsmodus unter-

stützt. (Sensoren und Kraftbegrenzer verbaut) Für eine detaillierte Beschreibung der beschriebenen kollaborativen Methoden verweisen wir den interessierten Leser auf (ISO/TS15066 2016).

Auch wenn die Methode der Kraft- und Leistungsbegrenzung die flexibelsten kollaborativen Szenarien zulässt (z.B. indem Roboter und Menschen in unmittelbarer Nähe arbeiten und sogar Kontaktkräfte auf möglicherweise nutzbringende Weise austauschen), muss für jede Aufgabe Risikobewertungs- und -reduzierungsverfahren durchgeführt werden. Dieser Aspekt schränkt die Flexibilität des Robotersystems ein, den menschlichen Arbeitsablauf zu unterstützen, wenn sich die Aufgabe häufig ändert. Um für unseren Demonstrator ein Höchstmaß an Flexibilität zu erreichen, ist die Geschwindigkeits- und Abstandsüberwachung daher die geeignetste Betriebsart. Einzelheiten zur Umsetzung sind in Abschnitt 2.3 beschrieben.

Der Anwender beginnt mit dem Sortieren der Komponenten die ihm auf dem Förderband präsentiert werden. Währenddessen erfasst das RGBD-Kamerasystem Informationen über die vom Menschen entnommenen Bauteile und den zugehörigen Ablageort. Diese Informationen dienen als Trainingsbeispiele für das Robotersystem. Das Robotersystem prüft laufend für jedes Objekt, ob es eine Klassifizierung mit ausreichender Sicherheit durchführen kann, um dann den Menschen bei der Sortierung zu unterstützen. Dies benötigt für eine ausreichend hohe Anzahl von Demonstrationen, welche der Mensch für jede Art von Bauteil vorführt. Das Robotersystem, entsprechend seiner beabsichtigten Funktionalität, wird den Anwender zunehmend unterstützen und Arbeit abnehmen, während es weiterhin von ihm lernt.

Eine derartige Anwendung ermöglicht es uns die vorgeschlagene Kombination von kollaborativer Robotik, Machine Learning und Machine Vision-Konzepten für andere industrielle Anwendungen zu adaptieren, für die das Grundkonzept dieses Demonstrators ebenfalls verwendet werden kann (*mutatis mutandis*). Unter der Voraussetzung, dass die Sicherheit von Menschen die zusammen mit Robotern arbeiten immer gewährleistet wird, können menschliche Arbeitsabläufe erfasst und genutzt werden um die Roboter lernen zu lassen, wie sie im Laufe der Zeit immer häufiger unterstützend eingreifen können. Um lange Trainingsphasen zu vermeiden, welche in solchen Anwendungen die Gesamtflexibilität des Systems einschränken würden, ist es entscheidend, dass Lernalgorithmen eine genügend hohe Zuverlässigkeit erreichen können, auch wenn nur wenige Demonstrationsmöglichkeiten vorhanden sind.



Abbildung 1: Hardware-Aufbau.

## 2.2. Hardware

Wie in Abbildung 1 gezeigt, besteht der Demonstrator aus einem leichtgewichtigen Roboterarm mit 7 Freiheitsgraden und Drehmomentsensorik in den Gelenken (Panda-Arm von Franka Emika GmbH) (1), einem speziell angefertigten Förderband (2) und Vision-Systemen bestehend aus einer RGB- (1920x1200, 30 fps) (3) und einer RGBD-Kamera (Microsoft Kinect V2) (4) zur Erkennung und Verfolgung von Werkstücken auf dem Förderband und Menschen im Roboterarbeitsbereich. Dabei werden Steuerungskonzepte für eine sichere Mensch-Roboter-Kooperation und maschinelle Lernansätze für einen benutzerfreundlichen Wissenstransfer implementiert und verglichen. Wie auf den Bildern dargestellt, werden zwei Computer eingesetzt. Der Erste (Intel i7-6700 3,4 GHz, 16 GB RAM, dedizierte Nvidia Quadro GPU) (5), läuft unter Windows und erlaubt die Verwendung von modernen Computer Vision Libraries und gleichzeitiger CUDA-Beschleunigung. Der Zweite (Intel i5-M3200 2,4 GHz, 8 GB RAM) (6) verwendet Ubuntu 16.04 mit dem Robot Operating System (ROS) (Kinetic Kame). Die Details der gesamten Software-Architektur mit einer detaillierteren Beschreibung der einzelnen Module folgen im nächsten Unterabschnitt.

## 2.3. Software

Die für den Demonstrator entwickelte Software besteht aus verschiedenen logischen Teilen (siehe Abbildung 2). Aus Effizienzgründen sind einige davon in gemeinsamen Modulen implementiert.



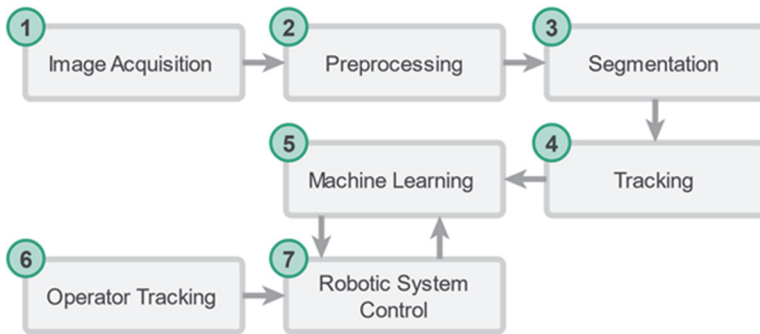


Abbildung 2: Struktur der Software.

### 1) Bilderfassung

Der Bilderfassungsschritt besteht aus einem Kamera-Interface-Modul (siehe Abbildung 3), welches eine Abstraktionsschicht für die verwendete Kamera-Hardware bereitstellt, und einem Resampling-Schritt, der die Bildauflösung auf die gewünschte Bildgröße reduziert. Es enthält einen Regelkreis, der eine niedrig aufgelöste Version des aktuellen Bildes verwendet, um die Belichtung abzuschätzen und die Belichtungsparameter der Kamera (Belichtungszeit und Verstärkung) im laufenden Betrieb anzupassen und die Gesamthelligkeit des Bildes konstant zu halten.

### 2) Vorverarbeitung

Die Vorverarbeitung besteht aus zwei Phasen. Zunächst wird eine adaptive Hintergrundsubtraktion durchgeführt, die das aufgenommene Bild in eine Vorder- und eine Hintergrundebene aufteilt. Dieses Teilmodul stellt auch Zugriff auf das Hintergrundmodell bereit. Der zweite Schritt führt auf die Vordergrundebene eine Kantenextraktion mit dem gängigen Canny-Algorithmus (Canny 1983) und einem morphologischen Operator durch und liefert als Ergebniss eine Edge-Map.

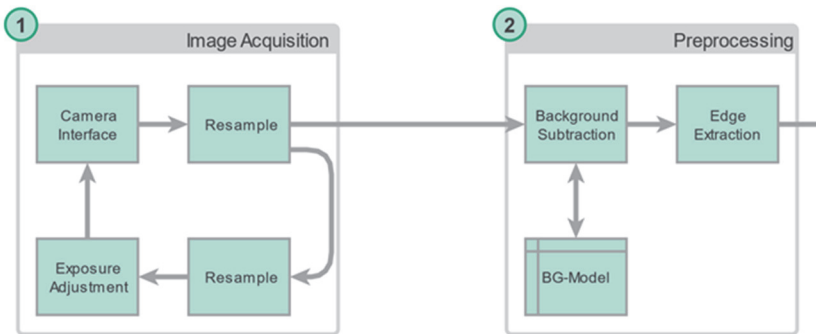


Abbildung 3: Bilderfassung und Vorverarbeitung.

### 3) Segmentierung

Der Segmentierungsschritt extrahiert die enthaltenen Konturen aus der durch den vorherigen Schritt erzeugten Edge-Map (siehe Abbildung 4). Die extrahierten Konturen werden dann angenähert, so dass nur die wichtigen Schlüsselpunkte erhalten bleiben. Die eingeschlossene Fläche innerhalb der Kontur entspricht dann einem erkannten Objekt und seine Merkmale werden extrahiert (z.B. Fläche, Schwerpunkt, Histogramm).

### 4) Tracking

Das Tracking-Modul übernimmt die aus dem aktuellen Frame extrahierten Objekte und vergleicht sie mit anderen kürzlich erkannten Objekten. Wenn eine passende Übereinstimmung gefunden wird, wird diese aktualisiert und die Trajektorie gespeichert. Wenn keine Übereinstimmung gefunden wird, wird die aktuelle Detektion als neues Objekt betrachtet und der Liste der getrackten Objekte hinzugefügt. Dieser Schritt erfolgt durch die Berechnung von Ähnlichkeitsmaßen zwischen den Objektmerkmalen. Die gefundenen und getrackten Objekte werden dann mit einem ZeroMQ<sup>1</sup> (ZMQ) basierten Publisher/Lazy-Subscriber-Modell an das ML-Modul gesendet. Dies ermöglicht es uns, das Lernsystem um mehrere parallele Methoden zu erweitern und die horizontale Skalierbarkeit durch die Verwendung mehrerer Workstations für die Berechnungen zu nutzen.

<sup>1</sup> <http://zeromq.org/>

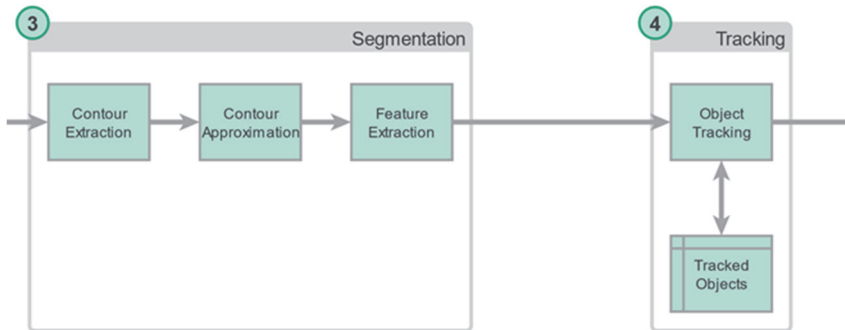


Abbildung 4: Segmentierung und Tracking.

## 5) Machine Learning

Das Machine Learning Modul (siehe Abbildung 5) besteht im Wesentlichen aus zwei Teilmodulen. Ein Lerner und ein Klassifikator. Dieser flexible Ansatz ermöglicht es uns, das Modul in einer Multi-Thread-Umgebung zu betreiben: Der Lerner aktualisiert das Modell, sobald neue Trainingsbeispiele eingehen, der Klassifikator verwendet dieses aktualisierte Modell, um nicht klassifizierte Beispiele zu kategorisieren. Jedes Submodul verfügt über eigene ZMQ-Sockets zur Kommunikation untereinander und mit den anderen Modulen des Systems.

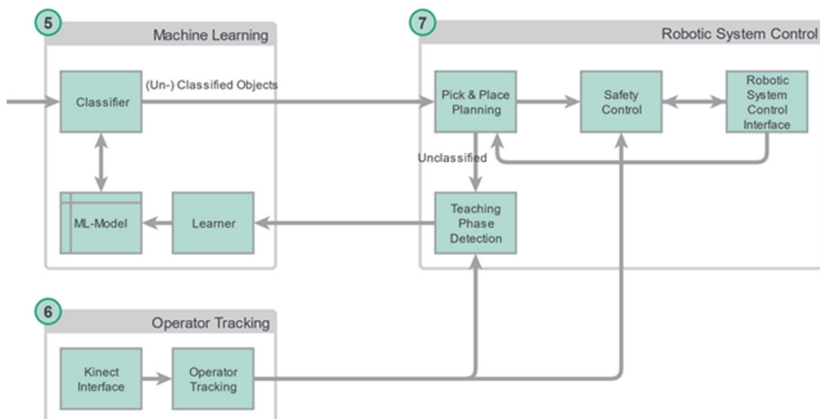


Abbildung 5: Machine Learning, Operator Tracking und Steuerung des robotischen Systems.

Das Lerner-Submodul wartet auf neue Trainingsbeispiele. Sobald ein neues Trainingsbeispiel eingegangen ist, wird es einem Stochastic Gradient Descent (SGD)<sup>2</sup> Klassifikator zugeführt, der uns die Möglichkeit gibt, iteratives Lernen ohne vorherige Schulung durchzuführen. Dies wird auch als Incremental-Learning oder Out-of-Core-Learning bezeichnet. Wir verwenden eine logistische Verlustfunktion, die unter anderem den Vorteil hat, dass die Vorhersagen unseres Modells probabilistisch interpretiert werden. Nachdem der Lerner das Modell aktualisiert hat, wird der Klassifikator veranlasst, noch nicht klassifizierte Bilder durch Anwendung des aktualisierten Modells neu zu bewerten (mehr dazu im nächsten Abschnitt).

Wie der Lerner hat auch das Klassifikator Submodul ein eigenes ZMQ-Socket, welches auf neue Objekte wartet. Nach dem Empfang neuer Bilddaten führt der Klassifikator eine Reihe von Vorarbeiten durch, bevor er versucht das Bild zu klassifizieren. Dazu gehört die Überprüfung des "Alters" des Bildes<sup>3</sup> und die Sicherstellung, dass das aktuelle Bild nicht zu sehr von dem abweicht, was das Lerner Modul bereits gesehen hat. Sobald ein Bild diese Vorabprüfungen bestanden hat, wird eine Vorhersage gemacht. Ist der Klassifikator von der Vorhersage überzeugt<sup>4</sup>, wird das Ergebnis über ein weiteres ZMQ-Socket ausgegeben. Andernfalls wird das Bild in die zuvor genannte Unklassifiziert-Liste aufgenommen.

Die Stärke unseres Aufbaus ist seine Flexibilität - neue Beispiele können in wenigen Iterationen eingelernt und mehrere ML-Algorithmen parallel verwendet werden. In Zukunft werden wir komplexere Objekte mit unterschiedlichen Formen und Eigenschaften klassifizieren. An diesem Punkt werden wir höchstwahrscheinlich auf leistungsfähigere maschinelle Lernalgorithmen zurückgreifen, die möglicherweise mit Techniken wie Boosting kombiniert werden (Schapire 1990).

## 6) Mitarbeiter-Tracking

Das Online-Tracking des Menschen ist sowohl für die Sicherheit als auch für das Lernen aus seinen Handlungen erforderlich. Dazu nutzen wir die Bibliotheken des Software Development Kits der Microsoft Kinect (C)

---

<sup>2</sup> Siehe:

[http://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.SGDClassifier.html](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html)

<sup>3</sup> Es besteht keine Notwendigkeit, "alte" Bilder zu klassifizieren, da sich das Förderband ständig bewegt und der Roboter es nicht mehr greifen kann.

<sup>4</sup> Mit anderen Worten: wir haben ausreichend hohe Wahrscheinlichkeit für die Vorhersage.

zur Skeletterkennung, Segmentierung und Tracking der Körperteile. Diese Daten werden ohne Vorverarbeitung über das verbreitete User-Datagramm-Protokoll (UDP) an das auf der Linux-Maschine laufende Robotersteuerungsmodul übertragen.

## 7) Steuerung des Robotischen Systems

Aus Effizienzgründen sind die Module 1-4 (vgl. Abbildung 2) als ein einziges C++-Programm implementiert, das auf der CUDA-beschleunigten Workstation läuft und zur Steigerung der Performance stark auf Multi-Threading setzt. Teil 5 ist in Python implementiert, um die umfassende Verfügbarkeit von Machine-Learning-Bibliotheken auszunutzen. Durch die Verwendung von ZMQ steht es uns allerdings frei zukünftige Module in beliebigen anderen Sprachen zu entwickeln und in das System zu integrieren.

Die Steuerung des Robotersystems (siehe Abbildung 5, Teil 7) ist auf einem dedizierten PC mit ROS implementiert. Diese Komponente enthält vier Module: Pick & Place Planung, Lehrphasenerkennung, Sicherheitssteuerung und die Schnittstelle zum Robotersystem zur Bewegungssteuerung. Das Pick-and-Place-Modul nimmt den Zustand des Robotersystems auf, um das Aufnehmen und Ablegen der Werkstücke vom Förderband in die richtigen Boxen zu planen. Diese Aktion ist für diejenigen Werkstücke durchzuführen, welche mit ausreichender Sicherheit klassifiziert wurden. Das Lehrphasenerkennungsmodul zielt darauf ab, ein vom Menschen initiiertes Lehrereignis zu erkennen und dem Lernmodul sukzessive die entsprechenden Informationen in Form von Trainingsbeispielen zur Verfügung zu stellen. Das Sicherheitskontrollmodul ist so implementiert, dass es die Sicherheitsanforderungen an kollaborative Arbeitsbereiche berücksichtigt. Dieses Modul überprüft online die Position des menschlichen Operators und die jeweils aktuelle Roboterkonfiguration, um den Roboter vor potentiellen Kollisionen während der aktuellen Bewegung sicher anzuhalten. Wir berechnen potentielle Kollisionen, indem wir Kapseln um den Roboter und den Menschen herum berücksichtigen. Kapseln werden in der Literatur auch als sog. sphere swept volumes bezeichnet und können als Minkowski-Summe aus einem Segment und einer Kugel betrachtet werden (Ericson 2004). Während der Radius der den menschlichen Körperteilen zugeordneten Kapselkugeln groß genug gewählt wird, um sie zu umschließen, wird der Radius der den Roboter umgebenden Kapselkugeln ausreichend groß gewählt, um den Anforderungen der (ISO/TS15066 2016) für den kollaborativen Modus der Geschwindigkeitsabstandsüberwachung zu entsprechen. Die Umsetzung einer Anwendung mit sicherheitsrelevanten

Anforderungen an die komplette Hard- und Software geht über den Zweck des Demonstrators hinaus und wird daher nicht berücksichtigt. Dies ermöglicht uns ein maximales Demonstrationspotenzial bei gleichzeitig niedrigen Entwicklungskosten zu erreichen.

### 3. Diskussion und aktueller Status

Der vorgestellte Demonstrator ermöglicht es uns, den potenziellen Nutzen der Kombination von kollaborativer Robotik, Machine Learning und Vision in der Industrie anhand einer beispielhaften und bewusst einfach gehaltenen Anwendung aufzuzeigen. In diesem Fall konnten wir durch diese Kombination das Potenzial eines Incremental-Learning-Robotersystems aufzeigen, welches flexibel in Bezug auf Änderungen der Aufgaben ist, selbst wenn diese kurzfristig erfolgen.

Die Software-Architektur und insbesondere der Machine Learning-Teil sind so flexibel, dass neue Algorithmen und Klassifikationsstrategien schnell integriert werden können und wir uns schnell an neue oder veränderte Verhältnisse anpassen können (z.B. neue Fallstudien von Partnern aus dem öffentlichen oder privaten Bereich). Ziel dieses Aufbaus war es, die Komplexität bewusst zu reduzieren, um schnell zu einer funktionierenden Lösung zu gelangen und dennoch grundlegende Funktionalität zu demonstrieren. Daher ergeben sich für uns mehrere Einschränkungen: Größe, Gewicht (aufgrund der Abmessungen des Aufbaus), Objektfrequenz (die Bewegungsgeschwindigkeit des Roboters ist begrenzt), Form und Material (manche Objektarten/-formen können vom installierten Greifer nicht aufgenommen werden), .... Einige dieser Beschränkungen können mit fortgeschritteneren Algorithmen angegangen werden, andere werden von der Hardware auferlegt und erfordern ein Upgrade, um sie zu beseitigen.

Zukünftige Arbeiten werden jedoch sowohl die Berücksichtigung von Objekten mit unterschiedlicher Form und Farbe als auch die Analyse der Robustheit in Bezug auf suboptimale Lichtverhältnisse beinhalten. Außerdem planen wir, verschiedene Methoden der Robotersteuerungs- und Sicherheitssteuerungsalgorithmen für die reaktive Bahnplanung zu evaluieren. Das Machine-Learning-System wird auch erweitert werden, um verschiedene Lernsysteme und Klassifikatoren miteinander zu vergleichen und ihre jeweilige Performance hinsichtlich z.B. Anlerndauer, Fehlerwahrscheinlichkeit, usw. zu untersuchen. Es ist des Weiteren geplant, das Robotersystem auf ROS2 zu migrieren, sobald dieses ausreichend ausgereift ist, um so die Echtzeit-Steuerungsmöglichkeiten der gesamten Anwendung zu verbessern.

Dieser Demonstrator konzentriert sich darauf, die Vorteile der Kombination von Mensch-Roboter-Kollaboration, Machine-Vision und -Learning zu demonstrieren. Für eine effektive Umsetzung des vorgeschlagenen Konzepts in KMUs sind spezielle Machbarkeitsstudien erforderlich, der Demonstrator kann hierbei aber bereits für den initialen Proof-of-Concept eingesetzt werden.

## Literatur

- Billard, A. G., Calinon, S., Dillmann, R. (2016). Handbook of Robotics 2nd Edition, ch. Learning from humans, S. 1995-2014. Berlin Heidelberg: Springer Verlag.
- Bouras B. A. (2016). Product Lifecycle Management for Digital Transformation of Industries. PLM. IFIP Advances in Information and Communication Technology, Vol. 492. Cham: Springer Verlag, S. 539-548.
- Canny, J. F. (1983). Finding edges and lines in images. MSc. Thesis. Boston: MIT.
- Ericson, C. (2004). Real-time collision detection. CRC Press, 2004.
- Hatvany, J., Nemes, L. (1978). Intelligent manufacturing systems— a tentative forecast, IFAC Proceedings Volumes, Vol. 11, (1), S. 895 – 899. 7th Triennial World Congress of the IFAC on A Link Between Science and Applications of Automatic Control, Helsinki, Finland.
- Giusti A., Zeestraten, M. J., Icer, E., Pereira, A., Caldwell, D. G., Calinon, S., Althoff, M. (2018). Flexible Automation Driven by Demonstration: Leveraging Strategies that Simplify Robotics, IEEE Robotics & Automation Magazine, Vol. 25 (2), S. 18-27.
- Haddadin, S., Croft, E. (2016). Physical human–robot interaction, Springer Handbook of Robotics, S. 1835–1874, Berlin Heidelberg: Springer Verlag.
- ISO/TS 15066 (2016): Robots and robotic devices - Collaborative robots, International Organization for Standardization, Geneva, Switzerland.
- Matt, D. T., Rauch, E., Riedl, M. (2018). Knowledge transfer and introduction of Industry 4.0 in SMEs – a five step methodology to introduce Industry 4.0. In: R. Brunet-Thornton und F. Martinez (Hrsg.): Analyzing the Impacts of Industry 4.0 in Modern Business Environments, S. 256-282. Hershey, PA: IGI Global.
- Mittal, S., Khan, M. A., Wuest, T. (2016). Smart Manufacturing: Characteristics and Technologies. In: R. Harik, L. Rivest, A. Bernard, E. Benoit, A. Bouras (Hrsg.): IFIP International Conference on Product Lifecycle Management, S. 539-548. Cham: Springer Verlag.
- Monostori, L. (2003). AI and machine learning techniques for managing complexity, changes and uncertainties in manufacturing, Engineering Applications of Artificial Intelligence, Vol. 16, (4), S. 277-291.

- Pereira A., Althoff, M. (2018). Overapproximative Human Arm Occupancy Prediction for Collision Avoidance, *IEEE Transactions on Automation Science and Engineering*, Vol. 15 (2), S. 818-831.
- Pham D. T., Afify, A. A. (2005). Machine-learning techniques and their applications in manufacturing, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 219 (5), S. 395-412.
- Robla-Gomez, S., Becerra, V. M., Llata, J. R., Gonzalez-Sarabia, E., Torre-Ferrero, C., Perez-Oria, J. (2017). Working together: a review on safe human- robot collaboration in industrial environments, *IEEE Access* 5 S. 26754–26773.
- Russell, S., Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ, USA: Prentice Hall Press.
- Schapire, R. E. (1990). The strength of weak learnability, *Machine Learning*, Vol. 5, S. 197–227.
- Sharp, M., Ak, R., Hedberg, T. (2018). A survey of the advancing use and development of machine learning in smart manufacturing. *Journal of Manufacturing Systems* (in press).
- Tao, F., Cheng, Y., Xu, L. D., Zhang, L., Li, B. H. (2014). Cciot-cmfg: Cloud computing and internet of things-based cloud manufacturing service system, *IEEE Transactions on Industrial Informatics*, Vol. 10, S. 1435-1442.
- Wuest, W., Weimer, D., Irgens, C., Thoben, K.-D. (2016). Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, Vol. 4 (1), S. 23-45.
- Xu, L. D., He, W., Li, S. (2014). Internet of things in industries: A survey, *IEEE Transactions on Industrial Informatics*, Vol. 10, S. 2233-2243.
- Zanchettin, A. M., Ceriani, N. M., Rocco, P., Ding, H., Matthias, B. (2016). Safety in human-robot collaborative manufacturing environments: Metrics and control, *IEEE Transactions on Automation Science and Engineering*, Vol. 13 (2), S. 882–893.
- Zanchettin, A. M., Croft, E., Ding, H., Li, M. (2018). Collaborative Robots in the Workplace [From the Guest Editors], *IEEE Robotics & Automation Magazine*, Vol. 25 (2), S. 16-17.