

Energieeffizientes eKanban-System mit autonomen Sensormodulen zur Füllstandsmessung und Reinforcement Learning zur Messintervallanpassung

Energy-efficient eKanban system with autonomous sensor modules for level measurement and reinforcement learning for measurement interval adaptation

Markus Kreutz¹
Abderrahim Ait Alla²
Michael Lütjen²
Michael Freitag^{1,2}

¹ Universität Bremen, Fachbereich Produktionstechnik, Bremen, Deutschland

² BIBA – Bremer Institut für Produktion und Logistik GmbH an der Universität Bremen, Deutschland

Trotz der Fortschritte der Digitalisierung in der Industrie finden bei der Bestandserfassung immer noch manuelle Trigger Einsatz, da bestehende Lösungen zur Automatisierung des Prozesses mit hohen Kosten und Integrationsaufwand verbunden sind. In diesem Beitrag wird ein Ansatz für die Lösung dieses Problems dargestellt, welcher kostengünstige, autonome Sensormodule für die Füllstandsmessung zur Grundlage hat. Die Messung erfolgt hierbei nicht in festgelegten Intervallen, sondern wird auf Basis der Entnahmeintervalle der zu messenden Ladungsträger sowie der aktuellen Auftragslage von einem Reinforcement Learning Ansatz dynamisch und intelligent getriggert. Die ersten Hardware-Prototypen für die Messung der Entnahmeintervalle sowie für die Sensormodule werden ebenfalls im Beitrag vorgestellt.

[Schlüsselwörter: e-Kanban, Bestandserfassung, Reinforcement Learning, Volumenmessung]

Despite the progress of digitalization in industry, manual triggers are still used for inventory measurement, as existing solutions for automating the process are associated with high costs and integration efforts. This paper presents an approach for solving this problem, which is based on cost-effective, autonomous sensor modules for fill level measurement. The measurement is not performed at fixed intervals, but is triggered dynamically and intelligently by a reinforcement learning approach based on the intervals in which contents are taken from the relevant load carriers and the current order situation. The first hardware prototypes for measuring the access to load carriers for content removal and for the sensor modules are also presented in the article.

[Keywords: e-Kanban, inventory control, reinforcement learning, volume measurement]

1 EINLEITUNG UND KONZEPTBESCHREIBUNG

Die Kanban-Steuerung ist ein weit verbreiteter Pull-Ansatz, um geringe Bestände in der Produktion zu realisieren. Im Rahmen der Digitalisierung löst die elektronische Kanban-Steuerung (e-Kanban) das traditionelle Kanban-System ab. Ein hoher Automatisierungsgrad des eingesetzten e-Kanban-Systems minimiert menschliche Fehler und Personalkosten. Hierzu gehört auch die Automatisierung der Meldung, wenn der Bestand eines Bauteils einen Schwellenwert unterschreitet, welcher dann Nachfüllaufträge auslöst. Daher ist eine automatisierte Bestandsmessung bzw. -Überwachung notwendiger Bestandteil eines vollautomatischen e-Kanban-Systems. Eine solche automatisierte Bestandsüberwachung für den Inhalt von Ladungsträgern in einer Produktionsanlage ist allerdings für viele KMU (kleine und mittlere Unternehmen) aufgrund hoher Kosten, eingeschränkter Funktionalität und hohem Integrationsaufwand noch nicht realisierbar.

Diese Probleme können durch die Realisierung von autonomen, kostengünstigen Sensormodulen zur Bestandüberwachung gelöst werden, welche durch Batteriebetrieb einen „Plug and Play“-Einsatz ermöglichen und somit eine hohe Mobilität und einfache Integration der Sensoren realisieren. Derzeit auf dem Markt verfügbare Lösungen erfüllen diese Anforderungen nicht. Um trotz des Batteriebetriebs eine hohe Betriebsdauer zu gewährleisten, bevor die Batterien getauscht werden müssen, wird der Energieverbrauch der Module optimiert. Hierzu wird die Füllstandsmessung nicht in festen Intervallen ausgeführt, sondern nur, wenn auch eine tatsächliche Veränderung des Füllstands zu erwarten ist. Diese Prognose wird von einem KI-basierten Lösung, nämlich einem Reinforcement Learning Ansatz, ermittelt, welche die aktuelle Auftragslage sowie laufende Betriebsdaten bezüglich der Entnahme von Inhalten aus den zu überwachenden Ladungsträgern analysiert. Hierzu ist eine weitere

Hardwarekomponente erforderlich, nämlich ein Sensor, um die Entnahme aus den Ladungsträgern zu ermitteln. Aus Gründen der Energieeffizienz empfiehlt sich hier ein Ansatz mittels PIR-Sensoren.

Dieser Beitrag baut auf unseren vorherigen Beitrag [KAL21] auf und weitet diesen aus. In diesem Beitrag wird daher die erste Version eines solchen Sensormoduls für die Füllstandsmessung vorgestellt, das als Benchmark für zukünftige Entwicklungen dienen soll. Es werden die Integrationsmethodik und die Implementierung des Sensormoduls beschrieben sowie eine Analyse hinsichtlich seiner Genauigkeit und seines Energieverbrauchs durchgeführt. In Erweiterung zu [KAL21] wird in diesem Beitrag die Entnahmerkennung vorgestellt, sowie eine tiefere Diskussion über den Füllstandsmessungsalgorithmus sowie weitere Ergebnisse, die in [KAL21] aufgrund der Seitenbeschränkung nicht mehr Platz fanden.

2 STAND DER FORSCHUNG

Ein Kanban-System ist ein visuelles Signalsystem, welches sogenannte Kanban-Karten verwendet, um den Nachschub der in der Produktion verwendeten Komponenten bei Bedarf auszulösen [RMS19]. Das elektronische Kanban-System (e-Kanban) ist eine Weiterentwicklung des klassischen Kanban-Systems, das digitale Technologien wie Barcodes, QR-Codes und Sensoren nutzt, um die traditionelle Kanban-Karte zu ersetzen [SSA20]. E-Kanban bietet mehrere Vorteile wie die Rückverfolgbarkeit in Echtzeit und die Sichtbarkeit von Materialbewegungen in den Produktions- und Fertigungssystemen. Falsche Eingaben und fehlende oder leere Behälter können automatisch erkannt werden und Materialengpässe können durch diese zusätzlichen Informationen minimiert werden [SUC18].

Für die Implementierung von e-Kanban geben MacKerron et al. einen allgemeinen Leitfaden vor, der aus acht Schritten besteht [MKK14]. Im Schritt "Dimensionierung von Kanban-Schleifen/Größen" betonen die Autoren die Bedeutung der Integration von virtuellen Kanban-Karten. Um die Rückverfolgbarkeitsproblematik von Rohstoffen anzugehen, konzipierten Costa et al. eine RFID-basierte Lösung [CDF17]. Durch RFID kann die Funktionalität der Kanban-Karten abgebildet werden, die die Sichtbarkeit der Rohstoffe verbessern. Hinsichtlich der Verwendung von Barcodes führten Sapry et al. Interviews mit Honda-Mitarbeitenden durch [SSA20]. Sie beschreiben den Einsatz eines Barcodesystems im Unternehmen für die Logistik, welches den Planungsprozess unterstützen und die Bewegungen der für die Produktion verwendeten Materialien überwachen soll. Jedes Mal, wenn eine Einheit produziert wird, scannt eine Arbeitskraft aus der Produktion den Barcode, und ein Signal wird an das Logistikteam gesendet. Raog et al. implementierten die Barcode-Technologie, um den manuellen Nachschubprozess

zu ersetzen [RMS19]. Durch das Scannen der digitalen Kanban-Karte wird der automatische Prozess der Materialbedarfsbestellung ausgelöst. Die in der Logistikabteilung tätige Führungskraft prüft die Bestellung, bevor der Auftrag endgültig ausgeführt wird.

Somit haben digitale Technologien zu Automatisierungen innerhalb der Lieferkette beigetragen, indem Funktionen wie die automatische Generierung, Erfassung und Versendung von Informationen von einer Station der Lieferkette zur anderen realisiert wurden [NMM20]. Trotz dieser Verbesserungen müssen die Arbeiter*innen den vom Kanban-System ausgelösten Nachschubauftrag immer noch prüfen und annehmen. Ein vollautomatischer Nachschubauftrag ist mit RFID oder Barcode nicht realisierbar, da die tatsächlichen Füllstände oder eine Anzahl produzierter Einheiten, insbesondere kleinerer Komponenten, nur qualitativ durch Arbeitskräfte gemessen bzw. geschätzt werden und daher fehleranfällig sind. Um menschliche Fehler zu reduzieren und Arbeitskräfte von repetitiven Aufgaben wie der Überprüfung der Korrektheit von Nachschubaufträgen zu entlasten, soll ein kostengünstiges, selbstlernendes e-Kanban-System entwickelt werden, das Sensormodule zur genauen Füllstandserfassung verwendet, die dann automatisch Nachschubaufträge auslösen können. In diesem Beitrag wird die erste Version eines solchen Sensormoduls vorgestellt, das als Benchmark für zukünftige Entwicklungen dienen soll.

Bestehende Arbeiten zur Füllstandsmessung befassen sich meist entweder mit der Messung von Flüssigkeitsfüllständen [HFB08, KFF19] oder der Füllstandsmessung von spezifischen Anwendungen wie z. B. industriellen Kugelmøhlen [HJZ09, YJX16]. Neben reinen Füllstandsmessungen können auch Untersuchungen zur Komponentenerkennung und -zählung [BTR20, BTR20b] zur Ableitung von Füllständen herangezogen werden. Die Einschränkung solcher Methoden ist, dass die Objekte sichtbar sein müssen, d.h. nur die erste Schicht der sichtbaren Objekte kann bei dieser Methode berücksichtigt werden. Im Gegensatz dazu erfordert unser Anwendungsfall, Füllstände für Ladungsträger mit mehreren Schichten gestapelter Komponenten erkennen zu können. Ein allgemeiner Ansatz zur Füllstandsmessung von Bauteilen, unabhängig davon, ob sie fest oder flüssig sind, konnte in der öffentlich zugänglichen wissenschaftlichen Literatur nicht gefunden werden.

3 BESCHREIBUNG DER HARDWAREKOMPONENTEN

3.1 SENSORMODUL ZUR FÜLLSTANDSMESSUNG

Das Sensormodul setzt sich im Wesentlichen aus einem optischen Sensor und einer Recheneinheit zusammen. Der optische Sensor ist auf den zu überwachenden Ladungsträger zu richten, während die Recheneinheit für die Berechnung des Füllstands aus den Input-Daten verantwortlich ist.

Als optischer Sensor wurde vorerst eine Intel RealSense D435i Tiefenkamera gewählt. Ein Entscheidungskriterium bei der Wahl des Sensors waren kompakte Maße und ein geringes Gewicht, um den Anforderungen des Plug and Plays zu genügen, welches dieser Sensor mit Maßen von 90x25x25 mm und einem Eigengewicht von ca. 75 Gramm erfüllt. Mit einem empfohlenen Messbereich von 0.3 bis 3 Metern, einer Messgenauigkeit von 2 % bei 2 Metern Abstand und einer Auflösung von 1280x720 Pixeln bei einer Framerate von 30 Hertz für das Tiefenbild erfüllt der Sensor auch die messtechnischen Anforderungen. Zudem verfügt diese Variante der Kamera auch über eine inertielle Messeinheit (IMU), mit der die Ausrichtung im Raum, also beispielsweise die Richtung der Gravitation, ermittelt werden kann. Die Kamera wird über einen USB-A Anschluss an den Rechner angeschlossen und ermöglicht so eine besonders einfache Installation. Es sollte hierbei ein USB 3.0-kompatibler Anschluss verwendet werden. Auch USB 2.0 wird unterstützt, allerdings können hier, bedingt durch die limitierte Datenübertragungsrate, bestimmte Kombinationen von Auflösung und Framerate nicht verwendet werden.

Als Recheneinheit wurde eine NVIDIA Jetson Nano mit 4 GB RAM Arbeitsspeicher verwendet. Im Gegensatz zu vergleichbaren Einplatinencomputern wie dem Raspberry Pi 3 besitzt die Jetson Nano eine stärkere Grafikkarte (GPU). Diese wurde gewählt, da in der nächsten Iteration des Sensormoduls auch Algorithmen für Bildverarbeitung und Machine Learning erprobt werden sollen und diese GPU-lastig arbeiten (siehe Abschnitt 7.1). Als Betriebssystem verwendet die Jetson Nano eine angepasste Ubuntu-Version. Als Vergleichsrechner für die Algorithmen wurde eine Desktop-PC mit einer AMD Ryzen 2700X CPU, 32GB RAM und einer NVIDIA 1060 GTX 6GB GPU herangezogen.

Für diesen Beitrag wurde das Sensormodul anhand von Kleinladungsträgern (KLTs) getestet. Hierzu wurde der Tiefensensor an einem handelsüblichen Kamerastativ befestigt, ungefähr zum Boden hin ausgerichtet und an die Jetson Nano angeschlossen. Die Jetson Nano bezieht ihren Strom direkt aus der Steckdose, während die Tiefenkamera über die USB-Verbindung mit Strom versorgt wird. Später soll die Stromversorgung per Akku erfolgen.

3.2 AUFBAU DER ENTNAHMEERKENNUNG

Abbildung 1 zeigt den Aufbau der Entnahmeerken- nung, welcher sich aus einem Mikrocontroller, dem ESP32, und einem PIR-Sensor zusammensetzt.

Im Gegensatz zur Jetson Nano hat die ESP32 eine geringere Rechenleistung und geringeren Stromverbrauch. Sie ist vergleichbar mit einem Arduino Board (<https://www.arduino.cc/>) und ist gut geeignet, um simple Bauteile anzusteuern. Die Programmierung für die ESP32 kann beispielsweise über den Arduino IDE (<https://www.arduino.cc/en/software>) erfolgen und wird in

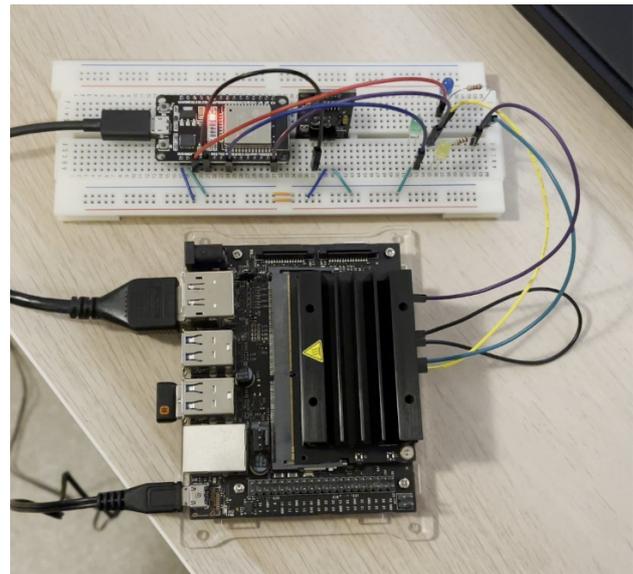


Abbildung 1: Entnahmeerken- nung mittels PIR-Sensor und ESP32, verbunden mit Jetson Nano, um diese bei erkannter Entnahme hochzufahren

einer C/C++-ähnlichen Sprache geschrieben. Anders als die Jetson Nano besitzt die ESP32 kein grafisches Betriebssystem. Der gespeicherte Code wird bei Stromzufuhr direkt ausgeführt. Auch muss die ESP32 nicht ‘heruntergefahren’ werden – stattdessen kann die Stromzufuhr einfach getrennt werden.

Ein PIR-Sensor ist ein passiver Infrarot-Sensor und gut geeignet, um Bewegungen von Menschen und Lebewesen zu erkennen. Durch die passive Messung der Infrarot-Strahlung ist der Sensor auch besonders energieeffizient.

Die beiden Komponenten wurden miteinander so verbunden und programmiert, dass die ESP32 mit Befehlen reagieren kann, sobald der PIR-Sensor eine bestimmte Anzahl von Malen ausgelöst wurde. Für diesen Beitrag sollte diese Entnahmeerken- nung die Jetson Nano hochfahren, sobald diese Anzahl erreicht wurde. An der Jetson Nano müssen hierfür zunächst die Pins ‘AUTO ON’ und ‘DISABLE’ überbrückt werden, damit die Einheit nicht automatisch hochfährt, sobald sie an Strom angeschlossen wird. Zudem sieht die Jetson Nano einen Pin ‘PWR BTN’ mit 5 Volt vor, welcher für den Anschluss eines physikalischen An-Aus-Knopfs vorgesehen ist. Sobald die Spannung an diesem Pin abfällt, fährt die Jetson Nano hoch. Dieser Pin wurde mit einem Widerstand an einem 3.3 Volt Pin der ESP32 angeschlossen, an dem per Code die Spannung zwischen an (HIGH) und aus (LOW) geschaltet werden kann. Dies kann genutzt werden, um den benötigten Spannungsabfall auf Abruf herbeizuführen. Somit war es möglich, die Jetson Nano über die Entnahmeerken- nungseinheit hochzufahren. Die Jetson Nano führt dann beispielsweise eine Füllstandsmessung durch, übermittelt die entsprechenden Daten, und fährt sich anschließend

selbst herunter, bis der nächste Trigger von der Entnahmeerkenkung kommt.

In Zukunft soll die Entnahmeerkenkung die Jetson Nano nicht mehr direkt hochfahren. Stattdessen sollen die Signale an einen zentralen Rechner mittels LoRa-Technologie übermittelt werden, welcher dann auf Grundlage eines Reinforcement Learning Modells entscheidet, wann die Jetson Nano hochzufahren ist.

4 EXPERIMENTELLER AUFBAU



Abbildung 2: Experimenteller Aufbau

Abbildung 2 zeigt den experimentellen Aufbau mit der Tiefenkamera.

Als Ladungsträger wurden vier Behälter mit unterschiedlichen Maßen verwendet, welche sich auch hinsichtlich ihrer Farbe unterschieden. Tabelle 1 beschreibt die ungefähren Dimensionen der verwendeten Behälter. Die Randdicke beschreibt dabei die minimale Randdicke, d.h. die Dicke des Rands an der dünnsten Stelle. Dieser Wert wird aufgeführt, da er für den Algorithmus für die Behältererkennung relevant ist.

Tabelle 1. Dimensionen der verwendeten Behälter.

Farbe des Behälters	Länge [cm]	Breite [cm]	Höhe [cm]	Dicke des Rands [mm]
Grau	40	30	27	15
Grün	43	36	23	21
Schwarz	36	27	15	7
Blau	25	13	9	1

Alle vier Behälter wurden für die Behältererkennung genutzt. Dieser Teil des Algorithmus identifiziert den Behälter im vorliegenden 3D-Bild.

Für die eigentliche Füllstandserkenkung wurden die zwei größten Behälter (grün und grau) verwendet. Diese entsprechen von den Maßen her der in etwa Größe von regulären Kleinladungsträgern (KLTs) (vgl. [VDA18], bezogen auf die Kurzbezeichnung 4280 – KLT mit der Grundfläche 400 x 300 mm und der Höhe 280 mm).

Als Füllmaterial wurden 28 Metallgussteile verwendet, welche in Abbildung 3 zu sehen sind. Die Metallgussteile sind zum Teil hohl und lassen sich ineinander stapeln.



Abbildung 3: Metallgussteile, die als Füllmaterial für die Ladungsträger für die Füllstandserkenkung verwendet wurden

5 BESCHREIBUNG DES FÜLLSTANDSMESSUNGS-ALGORITHMUS

Die Füllstandsmessung wurde in Python geschrieben, was den Vorteil hat, dass sie unabhängig vom Betriebssystem ausgeführt werden kann. Der Algorithmus für die Füllstandsmessung wird im Folgenden beschrieben:

1. Die Tiefenkamera liefert ein 2D-Tiefenbild, d.h. es liegt ein 2D-Bild vor, wo jeder Pixel einen Tiefenwert besitzt. Vor der eigentlichen Aufnahme des Bildes werden neun Frames übersprungen, da die Kamera sich an die Lichtverhältnisse anpasst und somit die ersten Frames für eine weitere Verarbeitung ungeeignet sind. Einige in der Bibliothek des Sensors verfügbare Vorverarbeitungsschritte, wie Glättung und Lochfüllung, werden auf das Tiefenbild angewendet. Abbildung 4 zeigt ein Beispiel für ein solches Tiefenbild.

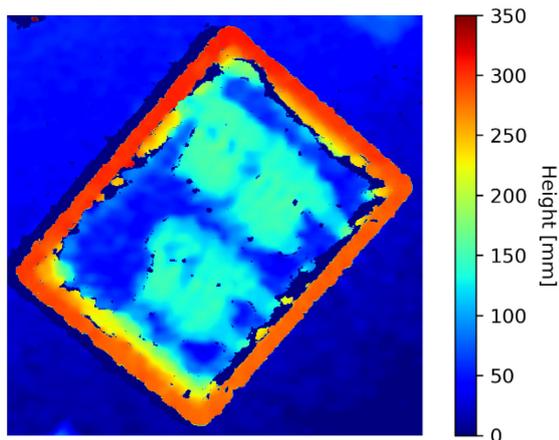


Abbildung 4: Beispielhaftes 2D-Tiefenbild eines zum Teil gefüllten Ladungsträgers (perspektivische Projektion).

2. Als nächstes wird das Tiefenbild in eine Punktwolke, also eine Sammlung von 3D-Punkten, konvertiert. Alle Punkte am Ursprung (0,0,0) werden aus der Punktwolke entfernt, da diese ungültige Punkte darstellen (zudem beträgt der Mindestabstand zur Kamera 0.3 m). Je nach Entfernung des Ladungsträgers zur Kamera wird im Vorhinein zudem auch ein maximaler Abstandswert für die Punkte zum (Kamera-) Ursprung als Parameter festgelegt. Liegt der Ladungsträger beispielsweise ca. 1 m unter der Kamera, könnte 2 m ein sinnvoller Wert für diesen Maximal-Wert sein. Alle Punkte mit größerem Abstand werden ebenfalls aus der Punktwolke entfernt.
3. Im nächsten Schritt wird die Bodenebene ermittelt. Dabei wird davon ausgegangen, dass die Kamera bereits grob auf den Boden ausgerichtet ist. Die Punkte werden in 5-cm-Bins entlang der z-Achse aufgeteilt. Der Bereich mit der größten Anzahl von Punkten wird ausgewählt. Auf diesen Punktesatz wird ein RANSAC plane fitting-Algorithmus angewendet, um die Bodenebene zu finden. Anschließend werden alle Punkte in der Punktwolke so rotiert, dass die neue z-Achse senkrecht zur gefundenen (Boden-)Ebene ausgerichtet ist.
4. Aus der gedrehten Punktwolke wird nun wieder ein geplättetes 2D-Tiefenbild erstellt. Die x- und y-Werte der Punkte werden in die Bildkoordinaten überführt und die z-Werte als die Werte an jedem Pixel dargestellt. Für diese Konvertierung wird ein Umrechnungsverhältnis zwischen Entfernung und Pixeln benötigt, welches als Parameter festgelegt wird. Im vorliegenden Fall zeigte sich ein Verhältnis von 400 Pixeln pro Meter als sinnvoller Wert. Zudem fallen bei dieser Konvertierung mehrere z-Werte pro Pixel an. Daher wird für jeden Pixel stets der höchste z-Wert aller diesem Pixel zugeordneten 3D-Punkte verwendet.

Im Gegensatz zum ursprünglichen 2D-Tiefenbild aus Schritt 1 besteht ein entscheidender Unterschied:

Während das von der Kamera aufgenommene Tiefenbild eine perspektivische Projektion der Szenerie zeigt, stellt das neu erzeugte, 2-dimensionale Tiefenbild eine orthografische Projektion dar, welche zur Bodenebene ausgerichtet ist. Vereinfacht lässt sich die perspektivische Projektion als ein aufgenommenes Bild einer punktförmigen Linse vorstellen, während bei der orthografischen Projektion die Linse sich als Ebene vorstellen lässt und somit sich die Projektion bei Translation parallel zur Bodenebene nicht ändert. Jeder Punkt bzw. Pixel in der orthografischen Projektion stellt genau den Abstand an diesem Punkt zur Bodenebene dar, weshalb beispielsweise Seitenwände von Objekten bei dieser Art der Projektion nicht auftauchen, da diese ein Effekt der Perspektive sind. Abbildung 5 veranschaulicht diesen Unterschied anhand von Beispielen.

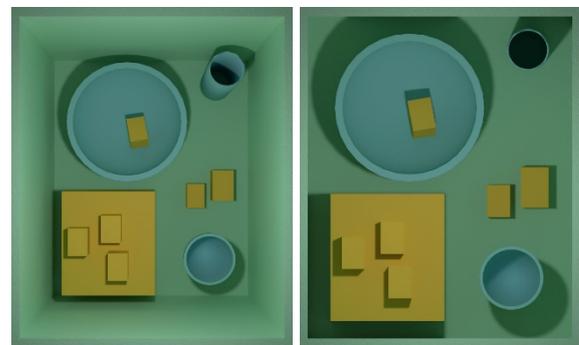


Abbildung 5: Beispiele für perspektivische Projektion (links) und orthogonale Projektion (rechts)

5. Im nächsten Schritt soll der Ladungsträger im Bild detektiert werden. Diese Detektion erfolgt anhand des oberen Randes des Ladungsträgers. Daher ist eben auch die Randdicke der Ladungsträger für den Algorithmus relevant (siehe Tabelle 1) und zudem muss dieser Rand auch im Bild gut sichtbar sein und die Inhalte des Ladungsträgers sollten nicht über dem Rand hinausragen. Um den Rand zu isolieren, wird aus dem erzeugten, 2-dimensionalen Tiefenbild ein binäres Bild erstellt, bei dem alle Punkte oberhalb und unterhalb der Containerhöhe (übermittelt als Parameter, beispielsweise 0.3 m) in einem Schwellenbereich (ebenfalls festgelegt als Parameter, beispielsweise 5 cm) auf 0 gesetzt werden und alle übrigen Punkte auf 1 gesetzt werden. Durch einen Connected-Component-Labeling-Algorithmus aus der scikit-image-Bibliothek angewandt auf das Binärbild werden zusammenhängende Bereiche im Bild ermittelt. Mit Hilfe der minAreaRect-Funktion von OpenCV wird ein minimales, rotiertes Rechteck an die gefundenen Bereiche gefittet. Passen die Maße zu den Seitenlängen des Ladungsträgers (Maße übermittelt als Parameter), so gilt der Ladungsträger als gefunden und wird entsprechend markiert.

6. Das Volumen des gefundenen Ladungsträger-Objekts wird berechnet. Die Höhe des Ladungsträgers wird aus den höchsten Punkten des Ladungsträger-Objekts bestimmt. Das Volumen wird dann durch Aufsummieren aller Pixelwerte im Ladungsträger-Objekt und Umrechnung der Fläche pro Pixel in Quadratmeter ermittelt. Auf diese Weise erhält man ein aktuelles Volumen und ein maximales Volumen. Wurde dem Algorithmus ein Leervolumen (Volumen des leeren Ladungsträgers) als Parameter übergeben, wird dieser Wert von beiden Werten subtrahiert. Das Dividieren der Werte ergibt den Füllstand in Prozent.

Anmerkungen zum Algorithmus:

- In Schritt 3 hätte auch die IMU der Tiefenkamera genutzt werden können, um die Punkte zur Bodenebene auszurichten. Im Praxistest war die IMU für diese Aufgabe allerdings wesentlich ungenauer als die beschriebene Methode.
- Schritt 3 stellt eine Einschränkung des Algorithmus dar, da hierdurch vorausgesetzt wird, dass erstens ein Bodenbereich in der Szene sichtbar sein muss, der nicht vollgestellt ist, und zweitens sich der Ladungsträger mit ihrem eigenen Boden auf dem Boden oder parallel dazu plaziert werden muss. Um das Problem zu umgehen, kann stattdessen auch die Kamera im Voraus senkrecht zum Ladungsträger ausgerichtet werden. Hier müssten Praxistests ermitteln, ob diese Lösung einfach umsetzbar ist oder ob es die Inbetriebnahme des Sensormoduls deutlich erschwert. Eine automatisierte Erkennung des Ladungsträgerrands in 3D ist auch denkbar, wäre aber deutlich ressourcenlastiger.
- Auch bezüglich des Rands des Ladungsträgers werden in Schritt 3 diverse Annahmen gemacht. So darf der Rand keine anderen Objekte in der Nähe von außerhalb des Ladungsträgers berühren. Zudem schlägt die Detektion fehl, wenn nicht der gesamte Rand erkannt werden kann. Auch wird vorausgesetzt, dass der gesamte Rand sich auf einer Höhe befindet. Einige dieser Annahmen können abgeschwächt werden, in dem nach dem Connected-Component-Labeling eine Art Clustering durchgeführt wird, bei der zusammenhängende Objekte zu neuen Objekten zusammengefasst werden. So könnte ein Rand, der im Tiefenbild als zwei Objekte auftaucht, dennoch zu einem einzigen Rand mit korrekten Maßen zusammengefasst werden. Auch denkbar wäre eine Linien-Detektion, um statt beliebiger Objekte Linien-Elemente zu finden, welche dann wieder zu einem kompletten Rand-Objekt zusammengefasst werden müssten. Diese Schritte würden allerdings auch die Rechenzeit erhöhen.
- In Schritt 5 wird aktuell die Ladungsträger-Detektion beendet, sobald ein Bereich / Objekt gefunden wurde,

dessen Maße den Ladungsträger-Maßen entsprechen. In Zukunft sollen alle Bereiche mit passenden Maßen berücksichtigt werden, sodass auch eine Füllstandserkennung für mehrere Ladungsträger gleichzeitig stattfinden kann. Denkbar ist auch, die Anzahl der zu überwachenden Ladungsträger vorzugeben. Hier muss dann, falls mehr passende Ladungsträger-Kandidaten im Bild gefunden werden als zu überwachende Ladungsträger, anhand von Fehlermaßen die besten Übereinstimmungen als die zu verwendenden Ladungsträger festgelegt werden.

6 ERGEBNISSE

6.1 DETEKTION DER LADUNGSTRÄGER

Der Algorithmus war in der Lage, alle in Tabelle 1 beschriebenen Ladungsträger zu erkennen. Für die graue und grüne Box wurde die Kamera in etwa 1 m Höhe über den Boxen positioniert. Bei der schwarzen Box musste die Kamerahöhe für eine positive Erkennung aufgrund des dünneren Boxenrandes auf etwa 0.7 m reduziert werden. Für die kleine blaue Box wurde die Kamerahöhe weiter auf ca. 45 cm vom Boden reduziert, um die Box mit dem Algorithmus erkennen zu können. Das ist bedingt durch die Fläche, die pro Pixel abgedeckt wird, welche mit der Höhe steigt, womit die effektive Auflösung mit der Höhe abnimmt. Für reguläre KLTs ist aber eine Höhe von 1m unbedenklich.

6.2 FÜLLSTANDSMESSUNG

Abbildung 3) Stück für Stück in die Ladungsträger gelegt. Hierfür wurden die grüne und die graue Box verwendet (siehe Tabelle 1), welche von den Dimensionen her in etwa dem von regulären KLTs entsprechen. Für jede Anzahl von Teilen im Ladungsträger wurden zehn Messungen durchgeführt. Die gemessenen Füllstände sind in Abbildung 6 dargestellt.

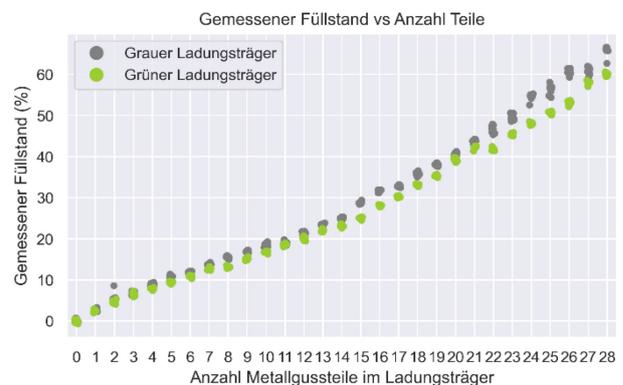


Abbildung 6: Gemessener Füllstand in Prozent aufgetragen gegen die Anzahl der Metallgussteile im Ladungsträger

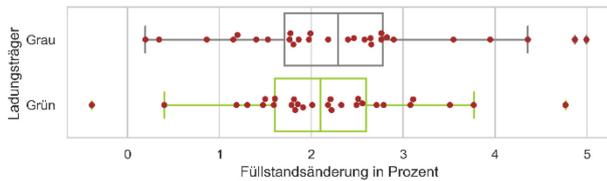


Abbildung 7: Verteilung der Füllstandsänderungen für Änderungen der Größe 1, d.h. bei Entnahme oder Hinzugabe eines einzelnen Metallgussteils

Der Füllstand steigt wie erwartet mit der Anzahl der Teile im Ladungsträger an. Die durchschnittliche Standardabweichung für die Füllstände betrug 0.5 % für die graue Box (für die grüne Box 0.3 %). Abbildung 7 zeigt die durchschnittliche Füllstandsänderung beim Hinzufügen bzw. Entfernen eines Teils aus dem Ladungsträger und zeigt eine durchschnittliche Änderung von etwa 2.3 % für jedes Teil im grauen Ladungsträger (2.1 % für die grüne Box). Unter der Annahme einer Gauß'schen Verteilung für die Füllstandsmessungen und einem Konfidenzintervall von 95 % können wir auf eine Füllstandsänderung von $2.3 \% \pm 1.0 \%$ pro Teil für die graue Box schließen (für die grüne Box $2.1 \% \pm 0.6 \%$). Diese Genauigkeit ist für den vorliegenden Anwendungsfall ausreichend, um die Änderung des Füllstands für die verwendeten Ladungsträger und die Teile zu verfolgen.

In Abbildung 6 fällt auf, dass der Algorithmus auch beim vollen Ladungsträger mit 28 Teilen (siehe Abbildung 3) einen maximalen Füllstandswert von etwa 65 % errechnet. Das liegt einerseits daran, dass auch wenn der Ladungsträger für das menschliche Auge voll erscheint, aufgrund von verbleibenden Leerräumen im Behälter das tatsächliche eingenommene Volumen von oben gesehen geringer ist als angenommen. Andererseits kann das auch daran liegen, dass aufgrund der Messgenauigkeit des Sensors die Höhenschätzung des Behälters mit Fehlern behaftet ist, was zu einer Füllstandsunterschätzung führen kann. Für den Anwendungsfall dieses Beitrags ist die Erkennung eines vollen Containers nicht so wichtig wie die Erkennung eines fast leeren Containers, daher ist dieses Phänomen nicht kritisch. Bei Bedarf kann, ähnlich wie bei der Registrierung des Leervolumens, ein Ladungsträger-Vollvolumen registriert werden, um die Ergebnisse entsprechend zu skalieren.

Weiter fällt auf, dass einige Füllstände zu unterschiedlicher Anzahl an Teilen fast gleich sind, beispielsweise beim grauen Ladungsträger für 26 und 27 Teile (siehe Abbildung 6). Dies ist auf die Stapelfähigkeit der Teile zurückzuführen. So können beispielsweise vier gestapelte Gussteile weniger Volumen einnehmen als drei gestapelte Teile. Hieraus lässt sich auch allgemein folgern, dass grundsätzlich zwar das eingenommene Volumen mit steigender Stückzahl steigt, dies aber nicht immer gilt.

6.3 ENERGIEVERBRAUCH UND AUSFÜHRUNGSZEIT

Um die Rechenanforderungen des in Abschnitt 5 beschriebenen Algorithmus zu ermitteln und die Rechenleistung zwischen Jetson Nano und Desktop-PC zu vergleichen, wurden 25 Messungen durchgeführt. Es wurden Messungen unabhängig von den Messungen für Abschnitt 6.2 durchgeführt, da für diesen Abschnitt noch zusätzlich RGB- und Tiefenbilder bei jeder Messung für mögliche spätere Untersuchungen gespeichert wurden, was die Ausführungszeit deutlich erhöht. Für die Messungen in diesem Abschnitt wird der Algorithmus nur mit den notwendigen Schritten für die Füllstandsmessung durchgeführt und speichert keine Bilder.

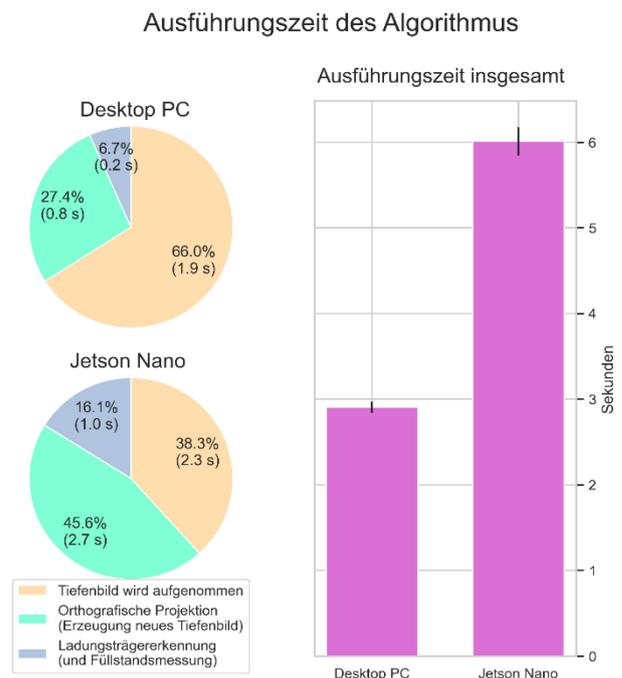


Abbildung 8: Rechts – Ausführungszeiten des Algorithmus für beide Systeme, unterteilt in die einzelnen Schritte des Algorithmus. Links – Gesamtausführungszeit des Algorithmus für beide Systeme mit 95% Konfidenzintervall.

Abbildung 8 zeigt die Resultate, wobei die Ausführungszeit des Algorithmus in die Abschnitte "Aufnahme des Tiefenbilds" (in etwa Schritt 1 in Abschnitt 5), "Erzeugung eines Tiefenbilds mit orthografischer Projektion" (Schritte 2-4) und "eigentliche Ladungsträgererkennung und Füllstandserkennung" (Schritte 5-6) unterteilt aufgezeigt wird. Beim letzten Schritt ist zu erwähnen, dass die Ausführungszeit hauptsächlich von der Ladungsträgererkennung dominiert wurde und die Füllstandsmessung vergleichsweise kaum Zeit benötigte.

Die Jetson Nano benötigte etwa die doppelte Zeit für eine Messung im Vergleich zum Desktop-PC. Die Zeit, bis das Tiefenbild aufgenommen wird, hängt größtenteils

von der Tiefenkamera ab und ist somit unabhängig vom System, auf dem der Algorithmus läuft. Wenn man diese ausschließt, brauchte die Jetson Nano etwa 3.7 Sekunden für den Algorithmus und der Desktop PC etwa 1.0 Sekunden, womit der Desktop-PC etwa viermal schneller war als die Jetson Nano. Da der Algorithmus in der aktuellen Form von der GPU nicht Gebrauch macht, erklärt sich der Unterschied hauptsächlich durch die stärkere CPU des Desktop-PCs.

Der Desktop-PC verbrauchte während der Messungen etwa 80 bis 90 Watt, während der Jetson Nano etwa 5 Watt verbrauchte. Daraus ergibt sich ein ungefährender Energieverbrauch von 8.3 mWh pro Messung für den Jetson Nano und 66.7 bis 75.0 mWh für den Desktop-PC. Somit ist die Jetson Nano trotz der höheren Ausführungszeit deutlich energieeffizienter unterwegs als der Desktop-PC und somit für den Anwendungsfall auch geeigneter. In Zukunft soll zudem untersucht werden, ob der Energieverbrauch weiter gesenkt werden kann, beispielsweise durch Untertaktung oder durch den Einsatz von Systemen mit noch geringerem Stromverbrauch.

7 DISKUSSION ZU ZUKÜNFTIGEN ENTWICKLUNGEN

7.1 KOSTENREDUZIERUNG

Das Ziel dieser Forschung ist die Entwicklung eines kostengünstigen und energiesparenden Sensormoduls. Die teuersten Teile des aktuellen Sensormoduls sind die Recheneinheit (Jetson Nano, \$99) und die Tiefenkamera (D435i, \$199). Der Algorithmus hat derzeit einen Spitzen-Speicherbedarf von (143 ± 1) MB. Daher sollte die Ausführung auf einem Raspberry Pi 3 Model A+ (\$25) mit ähnlicher CPU und 512 MB RAM möglich sein, da der Algorithmus die GPU derzeit nicht nutzt. Um den Energieverbrauch zu reduzieren, könnte die Recheneinheit untertaktet werden. Dies hat den Preis einer höheren Ausführungszeit. Das D435i kann durch das D435 (\$179) ersetzt werden, da der einzige Unterschied zwischen den beiden das Fehlen der IMU beim D435 ist, die der Algorithmus aktuell nicht nutzt. Die Verwendung des D415, ebenfalls von Intel, für \$149 könnte auch funktionieren, aber die Qualität der Messungen könnte darunter leiden.

Für die aktuelle Version des Sensormoduls wurden fertige Plug and Play Komponenten verwendet, beispielsweise die Intel Tiefenkamera, welche einfach per USB angeschlossen wird und für die eine fertige Bibliothek seitens des Herstellers zur Verfügung steht. Dies erhöht die Benutzbarkeit stark, aber steigert auch den Preis. Ähnlich wie bei der Entnahmeerkennung ist es denkbar, statt dieser fertigen Kameralösung einen Tiefensensor, der als Bauteil vorliegt, zu nutzen. Dieser müsste dann auch über ein Steckbrett an die Jetson Nano angeschlossen werden. Treiber und Kalibrierung würden bei einem solchen Bauteil voraussichtlich fehlen bzw. Nicht alle Funktionalitäten bieten, welche benötigt werden, weshalb diese

manuell durchgeführt werden müssten. Insgesamt würden so die Entwicklungskosten für das Sensormodul steigen, aber die Rohmaterialkosten sinken. Es gilt abzuschätzen, ob das lohnenswert wäre.

Eine weitere Möglichkeit, die Kosten für den Sensor zu reduzieren, wäre, den Tiefensensor durch eine gewöhnliche RGB-Kamera zu ersetzen. Dies würde einen völlig anderen Algorithmus erfordern. Es gibt Forschungsarbeiten zur Gewinnung eines Tiefenbildes aus einer Farbkamera mit Hilfe von Deep Learning [WMY19], wobei allerdings die Genauigkeit solcher Algorithmen gering ist und für den Anwendungsfall eventuell nicht ausreichen würden. Denkbar ist auch ein direkter Ansatz mit Deep Learning, bei dem Farbbilder als Input und der tatsächliche Füllstandswert als Output des Modells benutzt wird. Ein entsprechender Trainingsdatensatz lässt sich mithilfe des Sensormoduls, der noch die Tiefenkamera nutzt, aufbauen. Alternativ ist auch eine synthetische Generierung des Datensatzes über Computergrafiksimulation, beispielsweise mit Blender, denkbar. Diese Ansätze sollen in Zukunft verfolgt werden.

7.2 SENKUNG DES ENERGIEVERBRAUCHS

Wie in Abschnitt 3.2 bereits beschrieben wurde, sollen in Zukunft die detektierten Entnahmesignale aus dem Ladungsträger an ein Reinforcement Learning Modell weitergeleitet werden, welches dann dynamisch entscheidet, wann Messungen durchzuführen sind. So kann der Stromverbrauch, der für die Messungen benötigt wird, durch eine insgesamt geringere Anzahl an Messungen gesenkt werden. Auch soll untersucht werden, inwieweit sich der Stromverbrauch pro Messung noch weiter senken lässt, beispielsweise durch Untertaktung der Jetson Nano CPU.

7.3 INTEGRATION IN EIN E-KANBAN SYSTEM

Das geplante e-Kanban-System nutzt Methoden aus dem Data Mining und dem Reinforcement Learning, um die Bestände dynamisch anzupassen und zu minimieren. Nachschubaufträge werden nicht nur auf Basis aktueller Füllstände ausgelöst, sondern auch anhand von Faktoren wie Auftragsvolumen, Auftragspriorität, Produktionszeiten, Maschinenausfällen und Rüstzeiten. Rippel et al. zeigten die Machbarkeit und Leistungsfähigkeit eines solchen Systems in einer Simulationsstudie [RLT18]. Aufbauend auf dieser Forschung ist die Implementierung eines solchen e-Kanban-Systems in der realen Welt der nächste Schritt. Eine große Hürde bei diesem Schritt ist die Echtzeit-Datenerfassung für die Algorithmen. Der Beitrag dieser Arbeit stellt eine Lösung zur genauen Echtzeit-Datenerfassung der Füllstände von Ladungsträgern vor. Der nächste Schritt ist die Integration von Daten über Aufträge und Produktion. Die verwendeten Algorithmen benötigen Daten, um das Auftrags- und Produktionsverhalten des Unternehmens zu lernen. Der verwendete Reinforcement Learning Ansatz wird zunächst mit historischen

Trainingsdaten vortrainiert, um bereits bei Betriebsbeginn eine akzeptable Logik hinsichtlich der Messhäufigkeit der Füllstände zu besitzen. Anschließend wird das Modell im laufenden Betrieb mit in Echtzeit gesammelten Daten weitertrainiert, um die Zeitpunkte der Messungen weiter zu optimieren. Die Implementierung der erforderlichen Schnittstellen für diese Datenerfassung eine Hürde, die als nächstes angegangen werden muss.

8 ZUSAMMENFASSUNG

Ziel dieser Forschung ist die Entwicklung eines selbstlernenden e-Kanban-Systems. Dafür werden autonome, kostengünstige und energiearme Sensormodule zur Messung des Füllstands von kleinen und großen Ladungsträgern benötigt. In dieser Arbeit wird die erste Version eines solchen Sensormoduls unter Verwendung einer Intel Realsense D435i Tiefenkamera und eines NVIDIA Jetson Nano vorgestellt. In den Experimenten wurden zwei Behälter in der Größe von Kleinladungsträgern (KLTs) und 28 Metallgussteile als Inhalte zur Befüllung verwendet. Die Ausführung des Algorithmus dauerte 6 Sekunden, und die Messungen spiegelten die Änderungen durch Hinzufügen/Entfernen der Gussteile mit einer durchschnittlichen Änderung des Füllstands von jeweils $2.3\% \pm 1.0\%$ bzw. $2.1\% \pm 0.6\%$ für die Behälter wider. Auch wurde eine erste Version der Entnahmeerkenntnis vorgestellt, welche eine ESP32 und einen PIR-Sensor verwendet.

Da der aktuelle Preis des Sensormoduls noch recht hoch ist, wird in zukünftigen Forschungen der Einsatz einer Farbkamera anstelle einer Tiefenkamera untersucht. Dieser Ansatz wird wahrscheinlich die Entwicklung eines anderen Algorithmus erfordern.

9 DANKSAGUNG

Dieser Beitrag ist Teil des Projekts X-Kanban - Entwicklung eines selbstlernenden eKanban-Systems mit autonomen Sensormodulen, gefördert durch das Bundesministerium für Wirtschaft und Energie unter dem Förderkennzeichen "16KN076227".

LITERATUR

- [BTR20] Börold, Axel; Teucke, Michael; Rust, Johannes; Freitag, Michael: *Recognition of car parts in automotive supply chains by combining synthetically generated training data with classical and deep learning based image processing*. In: *Procedia CIRP*. 53rd CIRP Conference on Manufacturing Systems, Elsevier B.V, Amsterdam, NL, 2020. S. 377-382.
- [BTR20b] Börold, Axel; Teucke, Michael; Rust, Johannes; Freitag, Michael: *Deep Learning-based Object Recognition for Counting Car Components to Support Handling and Packing Processes in Automotive Supply Chains*. In: *IFAC-PapersOnLine*, Volume 53, Issue 2, 2020. S. 10645-10650.
- [CDF17] Costa, F.; do Sameiro Carvalho; M., Fernandes; J. M., Alves; A. C., & Silva, P: *Improving visibility using RFID—the case of a company in the automotive sector*. In: *Procedia Manufacturing*, Volume 13, 2017. S. 1261-1268.
- [HFB08] Holler, Gert; Fuchs, Anton; Brasseur, Georg: *Fill level measurement in a closed vessel by monitoring pressure variations due to thermodynamic equilibrium perturbation*. In: *2008 IEEE Instrumentation and Measurement Technology Conference*, 2008. S. 641-646, doi: 10.1109/IMTC.2008.4547115.
- [HJZ09] Huang, Peng; Jia, Min-ping; Zhong, Binglin: *Investigation on measuring the fill level of an industrial ball mill based on the vibration characteristics of the mill shell*. In: *Minerals Engineering*, 22(14), 2009. S. 1200-1208.
- [KFF19] Kandlbinder-Paret, Christoph; Fischerauer, Alice; Fischerauer, Gerhard: *Fill-level measurement using capacitance sensors with adaptively configurable electrodes for resolution enhancement*. In: *Measurement Science and Technology*, 30(4), 2019. S. 045101.
- [KAL21] Kreutz, Markus; Ait Alla, Abderrahim; Lütjen, Michael; Freitag, Michael: *Autonomous, low-cost sensor module for fill level measurement for a self-learning electronic Kanban system*. In: *Proceedings of 17th IFAC Symposium on Information Control Problems in Manufacturing (IFAC INCOM*

- 2021), Budapest, Hungary, 2021 [Accepted Paper].
- [MKK14] MacKerron, Grant; Kumar, Maneesh; Kumar, Vikas; Esain, Ann: *Supplier replenishment policy using e-Kanban: A framework for successful implementation*. In: *Production Planning & Control*, 25(2), 2014. S. 161-175.
- [NMM20] Núñez-Merino, Miguel; Maqueira-Marín, Juan Manuel; Moyano-Fuentes, José; Martínez-Jurado, Pedro José: *Information and digital technologies of Industry 4.0 and Lean supply chain management: a systematic literature review*. In: *International Journal of Production Research*, 58(16), 2020. S. 5034-5061.
- [RMS19] Raog, Koteswara; Michael, Lidwin Kenneth; Sriram, K. V.: *Optimization of inbound logistics by implementing e-kanban system in an automobile accessories manufacturing unit – a case study*. In: *Quality - Access to Success*, 20(170), 2019. S. 106-111.
- [RLT18] Rippel, Daniel; Lütjen, Michael; Theß, Michael; Freitag, Michael: *Application of Online Learning for the Dynamic Configuration of Kanban Systems*. In: 2018 Winter Simulation Conference (WSC), Gothenburg, Sweden, 2018. S. 3180-3192. DOI:10.1109/WSC.2018.8632274.
- [SSA20] Sapry, Hairul Rizad Md; Sabli, Siti Fairuz Hani Md; Ahmad, Abd Rahman: *EXPLORING E-KANBAN APPLICATION IN THE INVENTORY MANAGEMENT PROCESS*. In: *Journal of Critical Reviews*, 7(8), 2020. S. 6-10.
- [SUC18] Satoglu, Sule; Ustundag, Alp; Cevikcan, Emre; Durmusoglu, Mehmet Bulent: *Lean Production Systems for Industry 4.0*. In: *Industry 4.0: Managing The Digital Transformation*, Springer International Publishing, Cham, 2018. S.43-59. DOI:10.1007/978-3-319-57870-5_3
- [VDA18] VDA 4500 Kleinladungsträger (KLT-) System. VDA-Empfehlungen, 09.05.2018. Internet-Link: <https://www.vda.de/de/services/Publikationen/kleinladungstr-ger-klt-system.html>. Letzter Aufruf: 13.07.2021.
- [WMY19] Wofk, Diana; Ma, Fangchang; Yang, Tien-Ju; Karaman, Sertac; Sze, Vivienne: *Fast-Depth: Fast Monocular Depth Estimation on Embedded Systems*. In: 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019. S. 6101-6108. DOI:10.1109/ICRA.2019.879418
- [YJX16] Yan, Gaowei; Ji, Shanshan; Xie, Gang: *Soft sensor for ball mill fill level based on uncertainty reasoning of cloud model*. In: *Journal of Intelligent & Fuzzy Systems*, 30(3), 2016. S.1675-1689.
-
- Markus Kreutz, M.Sc.**, Research associate in the Faculty of Production Engineering of the University of Bremen.
- Address: BIBA - Bremer Institut für Produktion und Logistik, Hochschulring 20, 28359 Bremen, Germany, Phone: +49 421 218-50049, Fax: +49 421 218-98-50049, E-Mail: kre@biba.uni-bremen.de
- Dr.-Ing. Abderrahim Ait Alla**, Research associate at BIBA – Bremer Institut für Produktion und Logistik GmbH.
- Address: BIBA - Bremer Institut für Produktion und Logistik, Hochschulring 20, 28359 Bremen, Germany, Phone: +49 421 218-50082, Fax: +49 421 218-98-50082, E-Mail: ait@biba.uni-bremen.de
- Dr.-Ing. Michael Lütjen**, head of the department “Data Analytics and Process Optimization” at BIBA – Bremer Institut für Produktion und Logistik GmbH.
- Address: BIBA - Bremer Institut für Produktion und Logistik, Hochschulring 20, 28359 Bremen, Germany, Phone: +49 421 218-50123, Fax: +49 421 218-98-50123, E-Mail: ltj@biba.uni-bremen.de
- Prof. Dr.-Ing. Michael Freitag**, Director of BIBA - Bremer Institut für Produktion und Logistik and head of the research group Planning and Control of Production and Logistics Systems, Faculty of Production Engineering, University of Bremen
- Address: BIBA - Bremer Institut für Produktion und Logistik, Hochschulring 20, 28359 Bremen, Germany, Phone: +49 421 218-50001, Fax: +49 421 218-98-50001, E-Mail: fre@biba.uni-bremen.de