



Technische Hochschule Ostwestfalen-Lippe
Fachbereich 2
Medienproduktion

Bachelorarbeit zum Thema

**Entwicklung und Evaluierung eines multisensorischen Kamera-Setups zur
3D-Raumrekonstruktion in einer dynamischen Industrieumgebung mittels
Gaussian Splatting auf einem autonomen Robotersystem**

Sam Wiemann

Betreuender Professor: Dr. Ing. Alexander Kutter
Zweitprüfer: M.A. Colin Behrens
CC Lizenz: CC BY-SA (4.0)

Abstract

Die präzise und kosteneffiziente 3D-Rekonstruktion industrieller Umgebungen gewinnt zunehmend an Bedeutung, insbesondere für digitale Zwillinge und automatisierte Inspektionsprozesse. Diese Arbeit untersucht die Entwicklung und Evaluierung eines multisensorischen Kamera-Setups, das mit Gaussian Splatting eine speichereffiziente und schnelle Modellierung von Produktionsumgebungen ermöglicht. Dafür wurde ein mobiles, autonomes System auf Basis eines MiR-Roboters entwickelt, das Bilddaten von verschiedenen USB-Kameras (Logitech Brio, Anker C200), einer DSLR und iPhone Kamera sowie Raspberry Pi Kameras erfasst und verarbeitet.

Die zentrale Forschungsfrage adressiert die Effizienz, Bildqualität und Wirtschaftlichkeit von Gaussian Splatting im Vergleich zu etablierten Verfahren wie Photogrammetrie und LiDAR. Erste Testläufe mit DSLR-Aufnahmen und einer manuell rekonstruierten Pipeline zeigen, dass Gaussian Splatting hochdetaillierte Punktwolken generieren kann, jedoch mit Herausforderungen hinsichtlich Kameraausrichtung, Überlappung der Bilder und automatisierter Missionsplanung des Roboters konfrontiert ist.

Die Ergebnisse zeigen, dass das System eine schnelle und speichereffiziente 3D-Rekonstruktion ermöglicht, jedoch in seiner aktuellen Form noch Optimierungspotenzial aufweist. Besonders die genaue Kalibrierung der Kameras, eine verbesserte Synchronisation der Bilder sowie eine präzisere Steuerung des Roboters sind entscheidend für die Qualität der Punktwolken. Während Gaussian Splatting eine überzeugende Alternative zu klassischen Verfahren darstellt, ist die Systemstabilität derzeit noch abhängig von manuellen Eingriffen und Justierungen.

Zukünftige Arbeiten sollten sich darauf konzentrieren, die Automatisierung des gesamten Workflows weiter voranzutreiben, insbesondere durch den Einsatz verbesserter Kamera-Alignment-Algorithmen, Hardware-Trigger für synchrone Bildaufnahme und Machine-Learning-gestützte Bildoptimierung. Zudem könnte eine Erweiterung des Systems um zusätzliche Sensortechnologien wie Time-of-Flight-Sensoren oder LiDAR-Module helfen, die Detailgenauigkeit weiter zu verbessern und eine robuste Echtzeit-Rekonstruktion zu ermöglichen.

Schlagwörter: 3D-Rekonstruktion, Gaussian Splatting, Photogrammetrie, LiDAR, autonome Robotik, digitale Zwillinge, Industrie 4.0



Inhaltsverzeichnis

I	Abbildungsverzeichnis	5
II	Tabellenverzeichnis	5
III	Glossar	6
IV	Abkürzungsverzeichnis	8
1	Einleitung	9
1.1	<i>Problemstellung und Relevanz der Arbeit</i>	9
1.2	<i>Zielgruppe und Anwendungsbereiche</i>	10
2	Forschungsfrage und Hypothesen	11
2.1	<i>Forschungsfrage: Effizienz und Qualität kostengünstiger 3D-Rekonstruktion</i>	11
2.2	<i>Hypothesen zur Praxistauglichkeit von Gaussian Splatting</i>	11
3	Zielsetzung der Arbeit	13
3.1	<i>Entwicklung eines kostengünstigen 3D-Raumrekonstruktionssystems</i>	13
3.2	<i>Vergleich mit bestehenden Technologien</i>	13
3.3	<i>Evaluierung von Qualität, Kosten und Rechenzeit</i>	13
4	Stand der Forschung	14
4.1	<i>Überblick über bestehende 3D-Rekonstruktionsmethoden</i>	14
5	Material und Methodik	21
5.1	<i>Hardware</i>	21
5.1.1	<i>Kamerasysteme</i>	21
5.1.2	<i>Roboterplattform</i>	22
5.1.3	<i>Computerhardware für Berechnungen</i>	22
5.1.4	<i>Raspberry Pi 4</i>	22
5.2	<i>Software</i>	22
5.3	<i>Versuchsaufbau</i>	23
5.3.1	<i>Testumgebung und Kamera-Setups</i>	23
5.4	<i>Vergleichskriterien für Evaluierung</i>	25
6	Mögliche Anwendungsfälle	26
6.1	<i>Digitale Zwillinge für Produktionsumgebungen</i>	26
6.2	<i>Automatisierte Inventarisierung im Einzelhandel</i>	26
6.3	<i>Virtuelle Museen und Denkmalschutz</i>	27
6.4	<i>Einsatz im Bauwesen und Architektur</i>	27
6.5	<i>Erweiterte Anwendungen in der Industrie 4.0</i>	27
7	Praktische Umsetzung und Herausforderungen	29
7.1	<i>FTP Server und Fernzugriff</i>	29
7.2	<i>Erweiterung des MIR für eine USB-C Schnittstelle</i>	29
7.3	<i>Entwicklung der Missionsplanung für den Roboter</i>	30
7.3.1	<i>Erster Ansatz: Steuerung über das Web-Interface des MiR</i>	30

7.3.2	Zweiter Ansatz: Steuerung über API mit automatisierter Wegpunkt-Erstellung	31
7.3.3	Finaler Ansatz: Strategische Punkte und dynamische Wegpunkte	32
7.4	<i>Synchronisation der Kameras und Bildaufnahme</i>	32
7.4	<i>Probleme mit API-Kommunikation und Hotfixes</i>	32
7.5	<i>Bildqualität und USB-Webcam Probleme</i>	33
7.5.1	Lösungsansätze zur Verbesserung der Bildqualität	33
7.6	<i>Camera Alignments</i>	34
7.7	<i>Sensoren und Protective Fields</i>	34
7.8	<i>Optimierungsmöglichkeiten durch maschinelles Lernen</i>	35
9	Diskussion und Ausblick	36
9.1	<i>Interpretation der Ergebnisse</i>	36
9.1.1	Fotos mit automatischem und manuellem (festem) Weißabgleich	42
9.2.1	<i>Kosten Vergleich mit anderen Systemen</i>	44
9.2.2	<i>Vergleich mit kommerziellen 3D-Scannern</i>	45
9.3	<i>Beantwortung der Forschungsfragen</i>	46
9.4	<i>Möglichkeiten zur Optimierung des Workflows</i>	46
9.5	<i>Weiterentwicklung durch neuronale Netze und Deep Learning</i>	48
9.6	<i>Zukunftsperspektiven für Gaussian Splatting in der Industrie</i>	48
9.7	<i>Skalierbarkeit und Erweiterungsmöglichkeiten</i>	48
9.7.1	Integration in andere 3D-Programme und Pipelines	49
9.7.2	Handheld- und mobile Anwendungen	50
9.8	<i>Fazit und abschließende Bewertung</i>	50
V	Literaturverzeichnis	52
VI	Anhang	56
VII	Eidesstattliche Erklärung	61

I **Abbildungsverzeichnis**

Abbildung 1: Raspberry Pi mit "ArduCam Multi-Camera Adapter Board" und vier angeschlossenen Raspberry Pi Kameras und aufgeschraubten Linsen	23
Abbildung 2: Das finale Kamera Setup bestehend aus Raspberry Pi und vier Logitech Brio Kameras montiert auf einem Sockel und Stativ	24
Abbildung 3: Das zweite Kamera Setup bestehend aus Raspberry Pi und acht Anker PowerConf C200 Kameras montiert auf einem Sockel	24
Abbildung 4: Überblick über das gesamte System bestehend aus MiR, Kamera Setup sowie Macbook für den Fernzugriff	25
Abbildung 5: MiR mit abgeschraubter Bodenplatte sowie Hydrauliksystem für die Errichtung einer USB-C Schnittstelle	30
Abbildung 6: Grafisches Web-Interface des MiR mit gesamten Kartenausschnitt	31
Abbildung 7: Grafisches Web-Interface des MiR und die erstellten Wegpunkte im ca. 1m Abstand	31
Abbildung 8: MiR und rotem LED-Indikator für einen blockierten Sensor im Frontbereich	35
Abbildung 9: Kamera 2 der vier Logitech Brio und festem Weißabgleich	42
Abbildung 10: Kamera 1 der vier Logitech Brio und festem Weißabgleich	42
Abbildung 11: Kamera 3 der vier Logitech Brio und festem Weißabgleich	43
Abbildung 12: Kamera 4 der vier Logitech Brio und festem Weißabgleich	43
Abbildung 13: "generate_mission.sh"- Skript zur Erstellung der Missionswege des MiR	58

II **Tabellenverzeichnis**

Tabelle 1: Vergleich der Technologien für 3D-Raumrekonstruktion	20
Tabelle 2: Kosten Vergleich von den evaluierten Kamerasystem auf dem MiR	44
Tabelle 3: Kosten Vergleich mit einem evaluierten Kamerasystem zu anderen kommerziellen 3D-Scannern	45

III Glossar

API (Application Programming Interface)

Eine Schnittstelle, die es Softwareanwendungen ermöglicht, miteinander zu kommunizieren und Funktionen oder Daten auszutauschen, ohne die interne Implementierung offenzulegen.

Camera Alignments

Der Prozess der exakten Ausrichtung und Kalibrierung mehrerer Kameras, um eine kohärente und fehlerfreie 3D-Rekonstruktion zu gewährleisten.

CPU (Central Processing Unit)

Die zentrale Verarbeitungseinheit eines Computers, die allgemeine Berechnungen durchführt und Steuerungsaufgaben übernimmt.

DSLR-Kameras (Digital Single-Lens Reflex)

Digitale Spiegelreflexkameras mit Wechselobjektiven, die durch ihre großen Sensoren und manuelle Steuerungsmöglichkeiten besonders für professionelle Foto- und Videoaufnahmen geeignet sind.

Firmware

Eine spezielle Software, die direkt in Hardware-Komponenten wie Mikrocontroller, Kameras oder Router eingebettet ist und deren Grundfunktionen steuert. Sie kann oft durch Updates verbessert oder erweitert werden.

Floater und Artefakte

Ungewollte Fehler oder Störungen in digitalen Bildern oder 3D-Modellen, die beispielsweise durch unzureichende Überlappung von Bildern, algorithmische Ungenauigkeiten oder Sensorfehler entstehen.

FTP-Server (File Transfer Protocol Server)

Ein Server, der das Hochladen, Speichern und Herunterladen von Dateien über das File Transfer Protocol (FTP) ermöglicht, oft für den Datenaustausch in Netzwerken genutzt.

GPU (Graphics Processing Unit)

Eine speziell für parallele Berechnungen optimierte Recheneinheit, die besonders für grafikintensive Anwendungen, maschinelles Lernen und wissenschaftliche Berechnungen genutzt wird.

Industrie 4.0

Ein Konzept, das die Digitalisierung und Vernetzung industrieller Produktionsprozesse beschreibt. Es umfasst Automatisierung, das Internet der Dinge (IoT), KI-gestützte Analyse und cyber-physische Systeme zur Optimierung der Fertigung.

MiR (Mobile Industrial Robot)

Ein autonomer, mobiler Roboter, der in industriellen Umgebungen zur Automatisierung von Transport-, Logistik- oder Scanning-Aufgaben eingesetzt wird.

(Shell-)Skript

Ein ausführbares Skript, das in einer Kommandozeilenumgebung (z. B. Bash oder PowerShell) Befehle automatisch ausführt. Es wird häufig zur Automatisierung von Abläufen und Systemsteuerung verwendet.

Punktwolke

Eine Menge von 3D-Koordinatenpunkten, die ein gescanntes Objekt oder eine Umgebung repräsentieren und für die Erstellung digitaler Modelle genutzt werden.

Python (Programmiersprache)

Eine interpretierte, universell einsetzbare Programmiersprache mit einfacher Syntax, die besonders in den Bereichen Datenanalyse, Automatisierung, maschinelles Lernen und Softwareentwicklung weit verbreitet ist.

Raspberry Pi

Ein kostengünstiger, kreditkartengroßer Einplatinencomputer, der für verschiedene Anwendungen in Programmierung, Automatisierung und Robotik eingesetzt wird.

Smartfactory

Eine hochautomatisierte, digital vernetzte Produktionsumgebung, die durch IoT (Internet der Dinge), künstliche Intelligenz und Echtzeitanalyse eine flexible und effiziente Fertigung ermöglicht.

TSDF (Truncated Signed Distance Function)

Ein Verfahren zur volumetrischen Darstellung von 3D-Daten, das häufig in der Robotik zur Echtzeit-Kartierung und Umgebungserfassung eingesetzt wird.

Web-Interface

Eine grafische Benutzeroberfläche, die über einen Webbrowser zugänglich ist und die Steuerung sowie Interaktion mit einem System oder einer Anwendung ermöglicht. Sie erlaubt eine plattformunabhängige Nutzung ohne lokale Installation von Software.

IV Abkürzungsverzeichnis

API	Application Programming Interface
AR	Augmented Reality
CPU	Central Processing Unit
DSLR	Digital Single-Lens Reflex
FTP	File Transfer Protocol Server
GPU	Graphics Processing Unit
IoT	Internet of Things
KMU	Kleine und mittlere Unternehmen
LiDAR	Light Detection and Ranging
MiR	Mobile Industrial Robot
ML	Machine Learning
MVS	Multi-View Stereo
NeRFs	Neural Radiance Fields
Pi	Raspberry Pi
PSNR	Peak Signal-to-Noise Ratio
RAM	Random Access Memory
SSH	Secure Shell
SSIM	Structural Similarity Index Measure
ToF	Time-of-Flight
TSDF	Truncated Signed Distance Function
VR	Virtual Reality

1 Einleitung

1.1 Problemstellung und Relevanz der Arbeit

Die industrielle Digitalisierung – welche sich unter dem Begriff „Industrie 4.0“ verbreitet hat – nutzt bahnbrechende Technologien zur digitalen Abbildung realer Produktionsanlagen. Aus den resultierenden virtuellen Modellen entstehen digitale Zwillinge sowie Simulationssysteme; sie ermöglichen autonome Navigationsprozesse. Die Abbildungen unterstützen vor allem Fertigungslinien mit automatisierten Abläufen sowie Logistikzentren wie auch Produktionsstätten. Sie bilden alles exakt nach und machen die Gegebenheiten durch dreidimensionale Darstellungen begreifbar. Die Verantwortlichen planen leichter voraus, überblicken sämtliche Abläufe und führen Wartungsarbeiten effizienter durch.

Die SmartFactoryOWL in Lemgo benötigt neue 3D-Modelle welche sich rasch anpassen lassen; Die Produktion und die Ausstellungen verändern sich permanent. Laufende Umbauten erfordern digitale Raumdarstellungen, welche sich unkompliziert aktualisieren lassen. Ein mobiler Roboter soll zukünftig mit Sensoren wie auch Kameras ausgestattet werden und präzise 3D-Modelle erfassen. Die Evaluation unterschiedlicher Aufnahmesysteme wie Raspberry-Pi-Kameras, DSLRs sowie USB-Webcams zeigt auf, welche Geräte sich für den Einsatz eignen.

Die Kosten und technische Komplexität von LiDAR oder kommerziellen 3D-Kamerasystemen übersteigen trotz ihrer hohen Präzision die finanziellen Möglichkeiten vieler kleiner und mittelständischer Unternehmen. Gaussian Splatting revolutioniert durch seine speichereffizienten Punktwolken-Darstellungen die dreidimensionale Bildverarbeitung; die Technologie optimiert den Speicherbedarf und beschleunigt die Datenverarbeitung besonders auf mobilen Geräten für Echtzeitanwendungen.

Das Hauptziel der Arbeit besteht in der Entwicklung eines Kamerasystems welches sich günstig einrichten lässt. Es nutzt USB-Webcams die überall im Handel verfügbar sind sowie Programme mit offengelegtem Quellcode. Die Lösung zeichnet sich durch ihren finanziellen Vorteil aus - sie passt sich vielseitig an unterschiedliche Bedürfnisse an und bewährt sich im praktischen Einsatz. Die Ausarbeitung analysiert die Eigenschaften der verwendeten Technik sowie betriebswirtschaftliche Gesichtspunkte. Zusätzlich erforscht sie wie stark das System in Bereichen wie Logistik, Bauwesen als auch dem Einzelhandel zum Einsatz kommen könnte.

Die 3D-Raumrekonstruktion gewinnt durch die industrielle Revolution 4.0 stetig an Bedeutung. Der Markt für digitale Zwillinge im Industriesektor wird bis 2032 jährlich um durchschnittlich 39,8 % zulegen [11], [12] – Betriebe mit digitalen Zwillingen im Einsatz, steigern dadurch ihren Umsatz um durchschnittlich 15% und verzeichnen 25% mehr Systemleistung durch z.B. weniger Ausfälle durch ungeplante Wartungen [11], [13]. LiDAR-Systeme übersteigen oft die finanziellen Möglichkeiten der Anwender und photogrammetrische Verfahren erfordern zu viele Arbeitsstunden. Gaussian Splatting eröffnet einen Ausweg, welcher Speicherplatz effektiv nutzt und eine schnelle Datenverarbeitung in Echtzeit ermöglicht.

Verglichen mit Matterport oder FARO - die mindestens 5.000€, aber häufig deutlich mehr als 10.000 Euro kosten - schafft das entwickelte System durch günstige Hardware und freie Software eine preiswerte Alternative für 3D-Aufnahmen [14], [15]. Wer will passt die Komponenten nach Bedarf an. Besonders kleinere Firmen profitieren von dieser Bauweise die sich ihren begrenzten Mitteln anpasst. Der gesamte 3D-Laserscanning-Markt wird laut Prognosen bis 2035 auf 4,214 Milliarden US-Dollar expandieren und wächst jährlich um 6,6% [16].

1.2 Zielgruppe und Anwendungsbereiche

Der frei zugängliche Quellcode lädt sowohl Entwickler als auch Forscher zur Mitarbeit ein. Die Technik erlaubt durch ihre Offenheit und die damit verbundenen knappen Ausgaben einen wertvollen Einsatz für Organisationen, die sparsam haushalten oder ihre Mittel anders verteilen müssen. Besonders Studenten sowie kleine Bildungseinrichtungen und Einzelpersonen ziehen daraus Nutzen. Das Auswählen der Kameras und die genaue Überprüfung aller notwendigen Voraussetzungen verschlingen beim Aufbau des Systems aktuell die meisten finanziellen Ressourcen.

Ein geplanter Entwicklungsschritt führt zum Umbau des Systems in eine handliche Version – sie bietet eine preiswerte Alternative zu den hochpreisigen Scan-Geräten am Markt. Die Bedienung gelingt sofort nach dem Anschließen und erfordert nur wenige Minuten Unterweisung; Technische Vorkenntnisse bleiben überflüssig. In der SmartFactoryOWL übernehmen Mitarbeitende (oder auch Studierende der TH OWL) die direkte Steuerung entweder vom Standort aus oder per VPN-Verbindung aus der Ferne. Ein zentraler FTP-Server speichert alle aufgenommenen Bilder und macht sie für spätere Bearbeitungsschritte verfügbar.

Die Berechnung der Gaussian Splats verlangt nach Computern mit erheblicher Rechenkraft. Ein Versuchsaufbau arbeitet mit einer GeForce RTX 4070 und einem AMD Ryzen 9 5950X, welcher 16 Kerne besitzt – dazu kommen 64 GB RAM unter Windows 11. Schnellere Komponenten beschleunigen die Rechenoperationen enorm; Ältere Systeme stoßen wegen ihrer begrenzten Vereinbarkeit mit Programmen wie Postshot an ihre Grenzen [17]

Ein aktuelles Bachelorprojekt von Fabian Merschel ergänzt das System durch eine Befehlszeilenlösung. Diese automatisiert die Berechnung von Punktwolken als auch Gaussian Splats. Anwendende erstellen einen Input- wie auch Output-Ordner und laden dann ihre Dateien hinein. Die komplette Bearbeitung könnte zukünftig auch über einen Server des KreativInstitut.OWL ablaufen. Studierende wie auch externe Partner können dann ihre Berechnungen starten und erhalten nach deren Abschluss eine Mitteilung per E-Mail. [20]

Die modulare Struktur lässt sich mühelos auf verschiedene Anwendungsbereiche abstimmen. Der offene Quellcode ermöglicht eigene Erweiterungen wie auch Optimierungen – daraus entsteht eine Grundlage für industrielle, kulturelle und wissenschaftliche Projekte; Sie nützt vor allem kleinen und mittleren Unternehmen sowie Institutionen, welche über begrenzte Mittel verfügen [19].

2 Forschungsfrage und Hypothesen

2.1 Forschungsfrage: Effizienz und Qualität kostengünstiger 3D-Rekonstruktion

Die folgende Analyse beschäftigt sich mit der Konstruktion eines portablen Kamerasystems für dreidimensionale Raumerfassung. Dafür werden handelsübliche USB-Webcams genutzt und an einem selbstfahrenden Roboter montiert, sowie das Programm entwickelt. Die Forschungsergebnisse werden nach ihrer Praxistauglichkeit und dem Verhältnis zwischen Aufwand und Nutzen – verglichen mit existierenden Marktprodukten – ausgewertet. Die Bewertung analysiert die Technikparameter und die Wirtschaftlichkeit und schafft einen Vergleich mit anderen 3D-Industriestandards.

Die Untersuchung stützt sich auf nachstehende Forschungsfragen:

- I. Wie realisiert man ein mobiles System zur 3D-Raumerfassung mit preisgünstigen USB-Webcams? Welche technischen Voraussetzungen erweisen sich als erfolgversprechend?
- II. Wie stellt sich das konzipierte System im Vergleich zu etablierten Marktlösungen bezüglich seiner Ergebnisse und seines Preises dar?
- III. Wo bieten sich Möglichkeiten zur Verfeinerung des Systems – vor allem bei der Abstimmung zwischen physischen Komponenten und Programmierung?
- IV. Auf welche Weise unterstützen die erfassten Informationen eine Optimierung der Fertigungsabläufe?
- V. Mit welchen praktischen Schwierigkeiten muss bei der Einführung eines derartigen Systems in sich ständig wandelnden Fertigungsanlagen gerechnet werden?

Die Hauptforschungsfrage vereint sämtliche Bestrebungen dieser Untersuchung:

Wie kann ein mobiles, kosteneffizientes Kamera-Setup für die präzise und speichereffiziente 3D-Rekonstruktion einer dynamischen Produktionsumgebung entwickelt und optimiert werden?

Diese Arbeit ist eine innovative Entwicklung und bereichert die Produktion mit leistungsstarken Werkzeugen; sie revolutioniert den Einsatz von 3D-Technologie im industriellen Umfeld. Die Technik, die sich durch Geschwindigkeit wie auch Präzision auszeichnet, verändert grundlegend die Prozesse der 3D-Modellierung und des 3D-Scannings. In den Fabriken erleichtert sie das Planen, Kontrollieren und Modifizieren der betrieblichen Abläufe.

2.2 Hypothesen zur Praxistauglichkeit von Gaussian Splatting

Die Hypothesen zur Praxistauglichkeit von Gaussian Splatting lassen sich auf mehrere Aspekte der Technologie beziehen. Erstens wird angenommen, dass **Gaussian Splatting eine effizientere Verarbeitung großer Punktwolken ermöglicht als klassische Photogrammetrie,**

insbesondere durch GPU-Beschleunigung und speicheroptimierte Datenstrukturen. Studien zeigen, dass Gaussian Splatting im Vergleich zu Multi-View-Stereo-Verfahren eine um bis zu 500 % schnellere Rekonstruktion ermöglicht [9].

Eine weitere Hypothese ist, dass die Punktwolkenqualität in Abhängigkeit von der Bildqualität und der Anzahl der Eingabebilder variiert. Da Gaussian Splatting auf Wahrscheinlichkeitsverteilungen basiert, wird vermutet, dass ungenaue oder verrauschte Eingabedaten stärkere Artefakte und eine reduzierte Detailwiedergabe erzeugen [21], [22].

Ein Vergleich zeigt die Stärken von Gaussian Splatting gegenüber LiDAR wie auch NeRF - besonders bei den Kosten und der nötigen Rechenleistung. LiDAR scannt zwar präziser; Gaussian Splatting bietet der Industrie aber einen kostengünstigeren Weg zur 3D-Erfassung [23].

3 Zielsetzung der Arbeit

Die vorliegende Bachelorarbeit verfolgt das ambitionierte Ziel, ein kosteneffizientes multisensorisches Kamera-Setup zu entwickeln und auf einem mobilen Roboter zu implementieren. Dieses System soll Bilddaten präzise erfassen, die zur Erstellung von Punktwolken genutzt und in Gaussian Splats umgewandelt werden können. Die Arbeit zielt darauf ab, eine Lösung zu schaffen, die sowohl technologisch innovativ als auch wirtschaftlich umsetzbar ist, um Anwendungen in dynamischen Produktionsumgebungen zu ermöglichen.

3.1 Entwicklung eines kostengünstigen 3D-Raumrekonstruktionssystems

Ein zentrales Ziel der Arbeit ist die Entwicklung eines 3D-Raumrekonstruktionssystems, das sich durch niedrige Kosten, Benutzerfreundlichkeit und Flexibilität auszeichnet. Um dies zu erreichen, werden handelsübliche Kamerasysteme wie Raspberry Pi Kameras, USB-Webcams, und als Vergleich eine kostspieligere DSLR-Kamera sowie die immer verfügbare Smartphone Kamera eines iPhone genutzt. Diese Hardwarekomponenten werden systematisch getestet und hinsichtlich ihrer Handhabung, Bildqualität und Kosten bewertet. Die Bilddaten werden anschließend durch eine speziell entwickelte Bildverarbeitungs pipeline verarbeitet, die die Konvertierung der Bilddaten in Punktwolken und deren Umwandlung in Gaussian Splats ermöglicht. Durch diese kosteneffiziente Kombination von Hardware und Software wird eine Lösung geschaffen, die auch für kleine und mittlere Unternehmen (KMU) erschwinglich ist.

3.2 Vergleich mit bestehenden Technologien

Zur Bewertung der Effektivität des entwickelten Systems wird ein Vergleich mit bestehenden Technologien wie Photogrammetrie und LiDAR durchgeführt. Dabei werden zentrale Aspekte wie Kosten, Bildqualität, Verarbeitungszeit und Flexibilität untersucht. Während LiDAR durch seine hohe Genauigkeit und Echtzeitfähigkeit hervorsteicht, ist die Methode oft mit erheblichen Kosten verbunden. Photogrammetrie hingegen erfordert zahlreiche Aufnahmen und hohe Rechenkapazität, bietet jedoch gute Ergebnisse bei mittleren Kosten. Gaussian Splatting hingegen kombiniert Effizienz und Kostenvorteile und ermöglicht eine speicherschonende Verarbeitung großer Datenmengen, was es zu einer vielversprechenden Alternative macht [9].

3.3 Evaluierung von Qualität, Kosten und Rechenzeit

Benchmarks und Tests auf moderner GPU-Hardware wie der NVIDIA RTX 4090 zeigen, dass Gaussian Splatting im Vergleich zu Photogrammetrie eine deutlich geringere Rechenzeit benötigt. Gleichzeitig bietet die Methode eine Punktdichte, die etwa 90 % der Genauigkeit eines LiDAR-Systems erreicht, jedoch zu einem Bruchteil der Kosten. [21] Diese Ergebnisse sollen dazu beitragen, die praktische Anwendbarkeit des Systems in dynamischen Produktionsumgebungen zu bewerten und Verbesserungspotenziale aufzuzeigen.

4 Stand der Forschung

4.1 Überblick über bestehende 3D-Rekonstruktionsmethoden

Die 3D-Raumrekonstruktion bezeichnet den Prozess, physische Umgebungen in digitale Modelle zu überführen. Dies geschieht durch die Aufnahme und Verarbeitung von Bild- oder Sensordaten, um eine digitale Punktwolke oder ein Mesh-Modell zu erzeugen. Die dafür eingesetzten Technologien basieren auf unterschiedlichen Prinzipien und zeichnen sich durch spezifische Stärken und Schwächen aus. Es gibt verschiedene Technologien, die hierbei Anwendung finden:

4.1.1 Photogrammetrie

Die Photogrammetrie ist eine Methode zur 3D-Rekonstruktion, die Fotografien aus unterschiedlichen Perspektiven verwendet, um räumliche Strukturen zu analysieren und darzustellen [24]. Algorithmen wie Structure from Motion (SfM) und Multi-View Stereo (MVS) nutzen dabei die Überlappungen zwischen den Aufnahmen, um Tiefeninformationen abzuleiten und dreidimensionale Modelle zu generieren [25]. Diese Methode liefert bei ausreichender Beleuchtung und qualitativ hochwertigen Bildern eine hohe Präzision. Die Möglichkeit, handelsübliche Kameras einzusetzen, macht die Photogrammetrie zu einer kosteneffizienten Lösung, die in Bereichen wie Architektur, Denkmalschutz und Filmindustrie weit verbreitet ist [24].

4.1.2 LiDAR

LiDAR (Light Detection and Ranging) ist eine Technologie, die Entfernungen mithilfe von Lasersignalen misst und millimetergenaue Punktwolken erzeugt. Sie zeichnet sich durch ihre hohe Präzision und Echtzeitfähigkeit aus, was sie ideal für den Einsatz in dynamischen Umgebungen wie autonomen Fahrzeugen, der Robotik und der industriellen Fertigung macht. Trotz ihrer etablierten Anwendung sind die hohen Anschaffungskosten und der erhebliche Wartungsaufwand wesentliche Einschränkungen. Insbesondere kleine und mittlere Unternehmen (KMU) sehen sich durch die teure Hardware und die oft notwendige aufwändige Kalibrierung vor Herausforderungen gestellt [26].

4.1.3 Neural Radiance Fields (NeRFs)

Neural Radiance Fields (NeRFs) sind ein modernes Verfahren zur hochdetaillierten 3D-Rekonstruktion und Darstellung von Szenen, das auf tiefen neuronalen Netzwerken basiert. NeRFs modellieren Szenen als kontinuierliche Funktion, die Farbe und Dichte an jedem Punkt im Raum beschreibt. Diese Funktion wird durch das Training des Netzwerks mit Bildern aus unterschiedlichen Perspektiven gelernt. Dabei werden physikalische Eigenschaften wie Licht, Farbe und Transparenz entlang von Strahlen (Rays) simuliert, um fotorealistische Darstellungen zu erzeugen [29].

Ein wesentliches Merkmal von NeRFs ist ihre Fähigkeit, realistische Beleuchtungseffekte und komplexe Geometrien präzise darzustellen. Insbesondere in statischen Szenen liefert die Methode eine hohe Detailgenauigkeit, was sie ideal für Anwendungen in der Film- und Spieleindustrie macht. Im Vergleich zu anderen Methoden wie Gaussian Splatting oder Photogrammetrie bietet NeRFs eine überlegene Darstellung von Licht- und Materialeffekten, allerdings zu einem hohen Rechenaufwand. Jede Szene muss individuell trainiert werden, was zu langen Verarbeitungszeiten führt [16], [17].

Benchmarks zeigen, dass NeRFs für hochkomplexe Szenen oder dynamische Objekte exzellente Ergebnisse liefern, jedoch signifikant mehr Rechenressourcen und Speicher benötigen. Dies macht sie weniger geeignet für Echtzeitanwendungen oder Szenarien, in denen große Umgebungen modelliert werden sollen. Trotz dieser Einschränkungen stellen NeRFs eine vielversprechende Ergänzung zu bestehenden Technologien dar, insbesondere für Anwendungsbereiche, die auf fotorealistic Renderings angewiesen sind. Optimierungen in Bezug auf Trainingszeit und Speicherbedarf sind jedoch notwendig, um die Einsatzmöglichkeiten zu erweitern [16] – [18].

4.1.4 Gaussian Splatting

Gaussian Splatting visualisiert Punktwolken durch eine Methode, die Datenpunkte als kontinuierliche Wahrscheinlichkeitsverteilungen auf Basis von Gauß-Kernen darstellt. In den 1990er-Jahren entstand dieses Verfahren, das Kerbl wie auch Kollegen 2023 durch schnellere Grafikkarten und verbesserte Algorithmen revolutionierten [9]. Die Technik brilliert durch ihre Sparsamkeit beim Speicherverbrauch und läuft in Echtzeit – sie verbraucht deutlich weniger Ressourcen als photogrammetrische oder triangulierende Verfahren. Gaussian Splatting erzeugt Bilder, die sich durch ihre Glätte wie auch Natürlichkeit auszeichnen; Selbst bei lückenhaften oder störungsbehafteten Eingabedaten entstehen überzeugende Resultate. Über diese Vorzüge etabliert sich das Verfahren in Robotersystemen, erweiterten Realitätsanwendungen und selbstfahrenden Fahrzeugen. Auch Standard-Computer reichen für Gaussian Splatting aus – eine preiswerte Alternative zu kostspieligen LiDAR-Systemen öffnet sich dadurch für eine breite Anwenderschaft [10], [19] – [23].

4.1.5 Weitere relevante Algorithmen

Zusätzlich zu den bereits beschriebenen Technologien ergänzen weitere Algorithmen die Möglichkeiten der 3D-Datenverarbeitung und -darstellung.

PointNet ist ein Deep-Learning-Ansatz, der Punktwolken direkt verarbeitet, ohne die Daten in Voxel oder Bilder umwandeln zu müssen. Dies führt zu einer erheblichen Reduktion der Rechenzeit und macht PointNet besonders geeignet für Anwendungen, bei denen Echtzeitanalysen und die direkte Interpretation von Punktdaten erforderlich sind [9].

Ein weiteres relevantes Verfahren ist die Poisson Surface Reconstruction, die geschlossene Oberflächen aus Punktwolken generiert. Diese Technik wird häufig für die Nachbearbeitung genutzt, um aus unstrukturierten Punktwolken vollständige und glatte Modelle zu erstellen [10].

Schließlich spielt die Truncated Signed Distance Function (TSDF) eine wichtige Rolle bei der volumetrischen Darstellung von 3D-Daten. TSDF wird insbesondere in der Robotik für Echtzeit-Kartierungen und die Darstellung komplexer Umgebungen eingesetzt, da es präzise Geometriedaten liefert und gleichzeitig eine effiziente Speicherverwaltung ermöglicht [11]. Diese Algorithmen erweitern das Spektrum der verfügbaren Technologien und tragen dazu bei, die Anforderungen in verschiedenen Anwendungsbereichen von Echtzeit-Navigation bis hin zur hochpräzisen Modellierung zu erfüllen.

4.2 Wissenschaftliche Studien

Gaussian Splatting hat sich als ein vielversprechender Ansatz für die effiziente und kostengünstige 3D-Raumrekonstruktion etabliert. Die Methode bietet eine speicherschonende und leistungsstarke Verarbeitung, die mit handelsüblicher Hardware wie Consumer-GPUs realisiert werden kann. Ein besonderes Highlight ist die Möglichkeit, kostengünstige Kamerasysteme wie DSLRs, Raspberry Pi-Kameras und sogar Smartphone-Kameras wie jene moderner iPhones zu nutzen. Diese mobilen Geräte, ausgestattet mit fortschrittlichen Kameratechnologien, liefern in Kombination mit der Gaussian-Splat-Technologie qualitativ hochwertige Ergebnisse und senken die Einstiegshürden erheblich. Dadurch ergeben sich neue Anwendungsfelder in Bereichen wie der Architektur, Denkmalpflege und industriellen Prozessüberwachung, die bislang auf teure Spezialhardware angewiesen waren [9].

Trotz der schnellen Fortschritte fehlt es bislang an umfassenden Literaturübersichten oder praxisorientierten Handbüchern, die sowohl wissenschaftliche als auch praktische Aspekte von Gaussian Splatting beleuchten. Die meisten Studien konzentrieren sich auf spezifische technische Details oder Anwendungen, während holistische Darstellungen rar sind. Die Methode befindet sich weiterhin in einem frühen Entwicklungsstadium, was ihre standardisierte Nutzung einschränkt. Gleichzeitig zeigt die wissenschaftliche Community großes Interesse: Plattformen wie Discord dienen als zentraler Austauschort für neue Entwicklungen, während regelmäßig Publikationen erscheinen, die theoretische und technologische Fortschritte dokumentieren [19] – [23].

Die Forschung zur 3D-Raumrekonstruktion zielt zunehmend auf die Entwicklung kostengünstiger und effizienter Lösungen ab, insbesondere durch die Nutzung von Consumer-Hardware und Open-Source-Software. Studien haben gezeigt, dass handelsübliche Kamerasysteme wie Raspberry Pi-Kameras oder DSLRs qualitativ hochwertige Ergebnisse liefern können. Auch Smartphones mit hochauflösenden Kameras und integrierter Sensorik haben sich als vielversprechend für 3D-Rekonstruktionsaufgaben erwiesen. Dies eröffnet neue Möglichkeiten für den Einsatz mobiler, leicht zugänglicher Plattformen in einer Vielzahl von Anwendungen [9], [19] – [23].

Ein weiterer Forschungsbereich befasst sich mit der Entwicklung multisensorischer Systeme, bei denen unterschiedliche Kamertypen und Sensoren kombiniert werden, um Genauigkeit und Effizienz zu optimieren. Insbesondere die Integration solcher Systeme auf mobilen

Plattformen wie autonomen Robotern bietet großes Potenzial, bleibt jedoch weitgehend unerforscht. Die technische Herausforderung liegt in der Synchronisation und Datenfusion der

Die Echtzeit-3D-Raumrekonstruktion hat aufgrund des steigenden Bedarfs an Mixed-Reality-Lösungen, digitalen Zwillingen und autonomen Navigationssystemen an Bedeutung gewonnen. Traditionelle hochpräzise Systeme wie LiDAR oder spezialisierte 3D-Kameras bieten zwar exakte Punktwolken, sind jedoch kostspielig und für viele kleinere Anwendungen nicht praktikabel. Gaussian Splatting stellt eine kostengünstige Alternative dar, die hohe Effizienz und Flexibilität bietet. Studien untersuchen zunehmend die Praxistauglichkeit kostengünstiger Kameras und Sensoren für industrielle Anwendungen sowie deren Optimierung hinsichtlich Genauigkeit und Rechenzeit [9], [27].

4.3 Einsatzmöglichkeiten von 3D-Scantechnologien

Die Einsatzgebiete von 3D-Scantechnologien sind vielfältig und decken ein breites Spektrum an industriellen, kulturellen und wissenschaftlichen Anwendungen ab. Sie ermöglichen sowohl die digitale Erfassung und Archivierung physischer Objekte als auch die Optimierung von Prozessen in dynamischen Umgebungen. Im Folgenden werden die wichtigsten Anwendungsbereiche erläutert:

Digitale Zwillinge:

In der Industrie dienen digitale Zwillinge zur Überwachung, Simulation und Optimierung von Produktionsprozessen. Sie ermöglichen präzise Analysen und Vorhersagen, wodurch Ausfallzeiten reduziert und Wartungsarbeiten effizient geplant werden können. Technologien wie Gaussian Splatting bieten hier eine kostengünstige Alternative zu teuren LiDAR-Systemen und adressieren speziell die Bedürfnisse kleiner und mittlerer Unternehmen (KMU), die oft über begrenzte Budgets verfügen [8].

Virtuelle Museen und Denkmalschutz:

Historische Artefakte und architektonische Stätten können mit Technologien wie Photogrammetrie oder Gaussian Splatting erfasst und digital konserviert werden. Diese Ansätze bieten eine zugängliche Möglichkeit, kulturelles Erbe zu bewahren und global zugänglich zu machen. Beispiele wie das Freilichtmuseum OWL, in dem ein historisches Dorf digitalisiert wurde, oder der Scan des Hermannsdenkmals zeigen das Potenzial dieser Technologien im Bereich der kulturellen Archivierung [4][10].

Autonome Navigation:

Für Anwendungen wie autonome Fahrzeuge und mobile Roboter spielt die Echtzeit-3D-Kartierung eine zentrale Rolle. LiDAR-Systeme sind aufgrund ihrer Präzision häufig die bevorzugte Wahl, jedoch kostenintensiv. Gaussian Splatting bietet eine vielversprechende Alternative, insbesondere für Szenarien mit weniger strengen Anforderungen an die Genauigkeit.

Diese Methode ist besonders vorteilhaft in budgetorientierten Projekten, in denen trotzdem eine hohe Verarbeitungsgeschwindigkeit erforderlich ist [5][8].

Lagerlogistik:

Die Automatisierung und Optimierung von Lagerprozessen, einschließlich der Bestandsaufnahme und der Überwachung von Lagerflächen, profitieren von 3D-Scantechnologien. Der Einsatz kostengünstiger Kamerasysteme, kombiniert mit modernen Algorithmen wie Gaussian Splatting, ermöglicht eine effiziente und präzise Überwachung der Lagerbestände. Dies trägt dazu bei, Prozesse zu optimieren und Fehler in der Logistik zu reduzieren [8].

Forschung und Praxisstudien:

Die bisherigen Anwendungen verdeutlichen das Potenzial von 3D-Scantechnologien, doch es besteht weiterhin Bedarf an zusätzlichen Studien, um die Praxistauglichkeit in verschiedenen Branchen zu bewerten. Insbesondere die Kombination neuer Algorithmen mit kostengünstiger Hardware könnte weitere Einsatzbereiche erschließen und die Technologie für eine breitere Zielgruppe verfügbar machen.

Diese Einsatzbereiche zeigen die Vielseitigkeit moderner 3D-Scantechnologien und unterstreichen deren Relevanz für Industrie, Kultur und Wissenschaft. Zukünftige Entwicklungen, insbesondere im Bereich kosteneffizienter Alternativen wie Gaussian Splatting, könnten die Akzeptanz und Anwendungsmöglichkeiten weiter steigern.

4.4 Vergleich der Technologien

Die Wahl der geeigneten Technologie für 3D-Raumrekonstruktionen hängt stark von den spezifischen Anforderungen ab, die je nach Anwendungsbereich variieren können. Zu den zentralen Kriterien zählen Kosten, Genauigkeit, Verarbeitungsgeschwindigkeit, Speicherbedarf und Flexibilität. Technologien wie Photogrammetrie, LiDAR, Gaussian Splatting und Neural Radiance Fields (NeRFs) zeichnen sich durch unterschiedliche Stärken und Schwächen aus, die sie für verschiedene Szenarien geeignet machen. Eine Übersicht über die wichtigsten Merkmale dieser Technologien bietet die Tabelle 1, die detaillierte Vergleiche ermöglicht.

Aus wissenschaftlicher Sicht bleibt LiDAR die Methode mit der höchsten Präzision, insbesondere für industrielle und wissenschaftliche Anwendungen, bei denen Genauigkeit im Submillimeter-Bereich erforderlich ist. Allerdings sind die hohen Kosten und der komplexe Wartungsaufwand wesentliche Einschränkungen, insbesondere für kleinere Unternehmen [6]. Photogrammetrie stellt eine solide Alternative dar, die kostengünstiger ist und eine hohe Flexibilität bei der Geräteauswahl bietet. Jedoch ist die Methode rechenintensiv und benötigt eine große Anzahl qualitativ hochwertiger Bilder, was die Verarbeitungszeit verlängert [7].

Gaussian Splatting hat sich als vielversprechende Technologie herausgestellt, die mit vergleichsweise einfacher Hardware wie Consumer-GPUs und handelsüblichen Kameras betrieben werden kann. Durch die Nutzung von Gauß-Kernen zur Repräsentation von Punktwolken

reduziert diese Methode den Speicherbedarf erheblich und ermöglicht eine Echtzeit-Verarbeitung, was sie besonders für Echtzeitanwendungen attraktiv macht. Benchmarks zeigen, dass Gaussian Splatting die Verarbeitung von Punktwolken signifikant beschleunigt. Für die Verarbeitung von 1 Million Punkten benötigt die Methode auf moderner Hardware etwa 0,8 Sekunden, im Vergleich zu 2 Minuten bei Photogrammetrie [8]. Dennoch ist die Qualität der Ergebnisse stark von der Eingabedatenqualität abhängig. Störungen wie verrauschte oder unscharfe Bilder können die Genauigkeit beeinträchtigen [4], [8].

Neural Radiance Fields (NeRFs) bieten eine hervorragende Detailgenauigkeit und realistische Beleuchtungseffekte, insbesondere bei statischen Szenen. Ihre Anwendung ist jedoch durch die hohen Rechenanforderungen begrenzt, da jede Szene individuell trainiert werden muss. Dies macht sie besonders für die Film- und Spieleindustrie geeignet, jedoch weniger praktikabel für Echtzeitanwendungen oder Szenarien mit dynamischen Umgebungen [6], [7].

Der bedeutendste Vorteil der Gaussian Splatting Methode ist der Zeitvorteil beziehungsweise die schnellere Verarbeitung der Eingabebilder, gerade in Referenz zu dem Ausgabeergebnis.

Eine umfassende Untersuchung zur Effizienz von 3D-Rekonstruktionsverfahren zeigt, dass Gaussian Splatting hinsichtlich der Rechenzeit deutliche Vorteile gegenüber herkömmlichen Methoden wie Photogrammetrie oder LiDAR-basierten Verfahren bietet. Studien belegen, dass 3D Gaussian Splatting die Verarbeitungszeit im Vergleich zu NeRF-basierten Methoden um das Fünffache reduziert, was insbesondere für die Innenraummodellierung von Vorteil ist [38]. Ergänzend dazu zeigen Untersuchungen, dass Gaussian Splatting in der Echtzeit-3D-Rekonstruktion eine doppelt so hohe Geschwindigkeit wie vergleichbare NeRF-basierte Methoden erreicht, bei gleichzeitig reduziertem Speicherverbrauch [39].

Darüber hinaus liefern weitere Arbeiten zusätzliche Belege für die Effizienz dieser Technik. So wird gezeigt, dass qualitativ hochwertige Rekonstruktionen in nur drei Minuten berechnet werden können – eine Verbesserung um den Faktor 12 im Vergleich zu herkömmlichen NeRF-Ansätzen [40]. Im Vergleich zu Photogrammetrie, die für die Verarbeitung großer Punktwolken mehrere Minuten benötigt, und LiDAR-basierten Verfahren, die spezialisierte und kostspielige Hardware erfordern, stellt Gaussian Splatting somit eine äußerst effiziente Alternative dar [14]. Diese Ergebnisse unterstreichen das Potenzial von Gaussian Splatting für Anwendungen, die eine schnelle und ressourcenschonende Punktwolkengenerierung erfordern.

Die folgende Tabelle fasst diese zentralen Eigenschaften und Unterschiede der Technologien zusammen und bietet eine Grundlage zur Auswahl der geeigneten Methode basierend auf den jeweiligen Anforderungen.

Tabelle 1: Vergleich der Technologien für 3D-Raumrekonstruktion [10]

Kriterium	Photogrammetrie	LiDAR	Gaussian Splatting	NeRFs
Kosten	Mittel	Hoch (> 10.000 €)	Gering (Consumer-Webcams möglich)	Mittel bis hoch (hohe Hardwareanforderungen)
Genauigkeit	Mittel bis hoch	Sehr hoch	Hoch, aber geringer als NeRFs (abhängig von Inputdaten)	Sehr hoch (bei statischen Szenen)
Verarbeitung	Langsam	Schnell, Echtzeitfähig	Sehr schnell (GPU-beschleunigt)	Sehr langsam (hohes Training erforderlich)
Speicherbedarf	Mittel	Hoch	Gering	Hoch
Flexibilität	Hoch	Niedrig	Sehr hoch	Mittel (statistische Szenen bevorzugt)
Einsatzgebiet	Architektur, Denkmalschutz, Filmindustrie	Wissenschaft, Industrie	Echtzeitanwendungen, Consumer-Bereich, AR/VR, Produkte die keine millimetergenauen Scans benötigen	Film- und Spieleindustrie

Der Vergleich dieser Verfahren macht deutlich: Eine Entscheidung für eine bestimmte Technik erfordert stets ein sorgfältiges Abwägen zwischen Nutzen und Aufwand und dem Einsatzzweck. LiDAR-Systeme erweisen sich für Industrieprozesse, bei denen jeder Millimeter stimmen muss, als unschlagbar – sie haben deshalb auch große Kosten. Die Photogrammetrie überzeugt durch ihre Anpassungsfähigkeit; sie verlangt nach leistungsstarken Computern als auch nach Aufnahmen, die gestochen scharf sein müssen. In der Gaussian-Splatting-Methode steckt enormes Potenzial für Projekte, bei denen absolute Präzision nicht im Vordergrund steht. Über NeRFs lässt sich sagen: Sie erzeugen faszinierende Bilder, funktionieren aber ausschließlich bei unbewegten Objekten und finden vor allem beim Film Verwendung.

5 Material und Methodik

5.1 Hardware

Die Wahl der Hardware beeinflusst maßgeblich die Effizienz und Skalierbarkeit von Systemen zur Punktwolkenverarbeitung. Leistungsstarke GPUs wie die NVIDIA GeForce RTX 4090 bieten im Vergleich zur RTX 4070 eine signifikant höhere Leistung. Benchmarks zeigen, dass die RTX 4090 in verschiedenen Anwendungen eine deutlich bessere Performance erzielt [41]. Der Arbeitsspeicher (RAM) spielt ebenfalls eine wichtige Rolle. Während die Geschwindigkeit des RAMs die Verarbeitungszeit beeinflussen kann, ermöglicht eine größere RAM-Kapazität die Verarbeitung umfangreicherer Projekte. Für speicherintensive Aufgaben wird daher empfohlen, mindestens 32 GB RAM zu verwenden [42]. Zusätzlich kann der Einsatz von NVMe-SSDs den Datenzugriff beschleunigen und somit den gesamten Workflow verbessern. Schnelle Speicherlösungen tragen dazu bei, Engpässe bei der Datenverarbeitung zu minimieren und die Effizienz des Systems zu steigern [42]. Durch die sorgfältige Auswahl und Kombination dieser Hardwarekomponenten lässt sich die Leistung bei der Punktwolkenverarbeitung erheblich optimieren.

5.1.1 Kamerasysteme

Die Raspberry Pi HQ Kamera mit einem 12-Megapixel-Sensor und einem 6mm Weitwinkelobjektiv bietet eine gute Bildqualität für DIY-Projekte. Allerdings gab es Komplikationen beim Einsatz des Arducam Multi-Camera-Adapters, der es ermöglichen sollte, mehrere Kameras an einem Raspberry Pi zu betreiben. Der Adapter ließ sich aufgrund von fehlerhaften Anleitungen und schlechten Tutorials nicht korrekt installieren. Mehrere Versuche, den Adapter mit verschiedenen Raspberry Pi Betriebssystemen zu nutzen, schlugen fehl. Das führte zu einem Wechsel von Raspberry Pi Kameras zu USB-Webcams, da der Multi-Camera-Adapter nicht zuverlässig funktionierte. Der Preis liegt bei etwa 60€ für Kamera und Objektiv [47].

Der Wechsel von Raspberry Pi Kameras zu USB-Webcams erfolgte aufgrund von Kompatibilitätsproblemen mit dem Arducam Multi-Camera-Adapter, der in Verbindung mit dem Raspberry Pi 4 (4GB) nicht zuverlässig funktionierte. Die Einrichtung der Software für das System erstreckte sich über mehrere Tage und wurde deshalb aufgegeben.

Die Anker PowerConf C200 bietet eine 2K-Auflösung, eine Bildrate von 30 FPS und ein Sichtfeld von 95°, was sie für Videokonferenzen gut geeignet macht. Sie lässt sich einfach anschließen und ist gut mit verschiedenen Betriebssystemen kompatibel. Ein Vorteil der Kamera ist das gute Preis-Leistungs-Verhältnis und die einfache Handhabung. Allerdings bietet sie im Vergleich zu 4K-Webcams eine geringere Bildqualität und begrenzte Anpassungsmöglichkeiten. Der Preis liegt zwischen 30€ und 60€, je nach Anbieter [43].

Die **Logitech Brio** bietet eine 4K-Auflösung und HDR, was sie besonders für professionelle Anwendungen wie Videokonferenzen und Streaming eignet. Mit einem flexiblen Sichtfeld von

90° bietet sie verschiedene Anpassungsmöglichkeiten. Der einzige Nachteil dieser Webcam ist der hohe Preis, der bei etwa 150 bis 200€ liegt. [44]

Als Vergleichskameras dient eine Nikon D7500 DSLR mit 4K Auflösung, sehr guter Qualität aber einem entsprechend hohen Anschaffungspreis von etwa 1000€, sowie eine Smartphone Kamera. Dafür wurde ein älteres, gängiges iPhone 12 Pro mit einer 12 MP Kamera verwendet. Anschaffungspreis ist hierfür circa 800€.

5.1.2 Roboterplattform

Der **MiR 600 EU Pallet Lift** ist eine autonome Lösung für das Handling von Paletten in industriellen Umgebungen, die mit dem **MiR 600 AMR** betrieben wird. Er ermöglicht das Transportieren von Paletten mit einer maximalen Nutzlast von 600 kg und erreicht eine Geschwindigkeit von 2 m/s. Mit einer Betriebsdauer von bis zu 10,5 Stunden und einem Hydrauliksystem für das Anheben und Senken von Paletten ist er für den effizienten Transport schwerer Lasten geeignet. Der Roboter nutzt fortschrittliche Sensortechnologie, darunter LiDAR und Kameras, für präzise Navigation und Hindernisvermeidung. Er lässt sich einfach in bestehende Layouts integrieren, benötigt keine Kabel, Magnete oder QR-Codes und kann über eine Web-Oberfläche, mobile App oder Fernbedienung gesteuert werden. Die MiR Fleet Management Software ermöglicht die zentrale Verwaltung und Optimierung mehrerer Roboter, während die API eine nahtlose Integration in bestehende Systeme ermöglicht [46].

5.1.3 Computerhardware für Berechnungen

Das System ist mit einem AMD Ryzen 9 5950X 16-Core Prozessor mit 3,40 GHz, 64 GB RAM und einer GeForce RTX 4070 Grafikkarte ausgestattet. Es läuft auf Windows 11 und nutzt ein 64-Bit-Betriebssystem. Dieses Setup ermöglicht eine effiziente Ausführung komplexer Anwendungen und Berechnungen.

5.1.4 Raspberry Pi 4

Der Raspberry Pi 4 Model B ist mit einem 1,5 GHz Quad-Core ARM Cortex-A72 Prozessor ausgestattet und verfügt über 4 GB RAM. Das System unterstützt WLAN und Bluetooth für drahtlose Kommunikation. Es wird ohne vorinstalliertes Betriebssystem geliefert, jedoch lässt es sich mit einem passenden OS wie Raspberry Pi OS betreiben.

5.2 Software

Um die Benutzerfreundlichkeit der Skripte zu erhöhen, könnten grafische Interfaces und kommentierte Tutorials implementiert werden. Darüber hinaus bietet maschinelles Lernen großes Potenzial für die Automatisierung und Verbesserung des Workflows. ML-Modelle könnten etwa Bildrauschen automatisch entfernen oder die optimale Kamera-Positionierung basierend auf Umgebungsdaten vorschlagen. Modelle wie DnCNN haben sich in der Bildrauschreduzierung als effektiv erwiesen und könnten eine sinnvolle Ergänzung zum bestehenden System darstellen [9], [38].

Postshot: Diese Software wurde zur Verarbeitung der aufgenommenen Bilder genutzt. Sie bietet Kommandozeilen-Befehle zur effizienten Verarbeitung von Bilddaten. Besondere Stärken liegen in der Vorverarbeitung und Filterung von Bildrauschen.

Python-Skripte: Zur Steuerung der Kameras und zur Übertragung der Daten auf einen FTP-Server wurden eigens entwickelte Python-Skripte verwendet. Diese Skripte ermöglichten die automatisierte Aufnahme synchronisierter Bilder von mehreren Kameras (siehe Anhang).

5.3 Versuchsaufbau

5.3.1 Testumgebung und Kamera-Setups



Abbildung 1: Raspberry Pi mit "ArduCam Multi-Camera Adapter Board" und vier angeschlossenen Raspberry Pi Kameras und aufgeschraubten Linsen [10]



Abbildung 2: Das finale Kamera Setup bestehend aus Raspberry Pi und vier Logitech Brio Kameras montiert auf einem Sockel und Stativ [10]



Abbildung 3: Das zweite Kamera Setup bestehend aus Raspberry Pi und acht Anker PowerConf C200 Kameras montiert auf einem Sockel [10]

Das finale Setup bestand aus vier oder acht USB-Webcams, die auf einem autonomen Roboter installiert wurden. Der Roboter navigierte autonom durch die SmartFactoryOWL und nahm dabei synchronisierte Bilder auf. Diese Bilder wurden auf einem zentralen FTP-Server gespeichert und anschließend mit Gaussian Splatting verarbeitet.

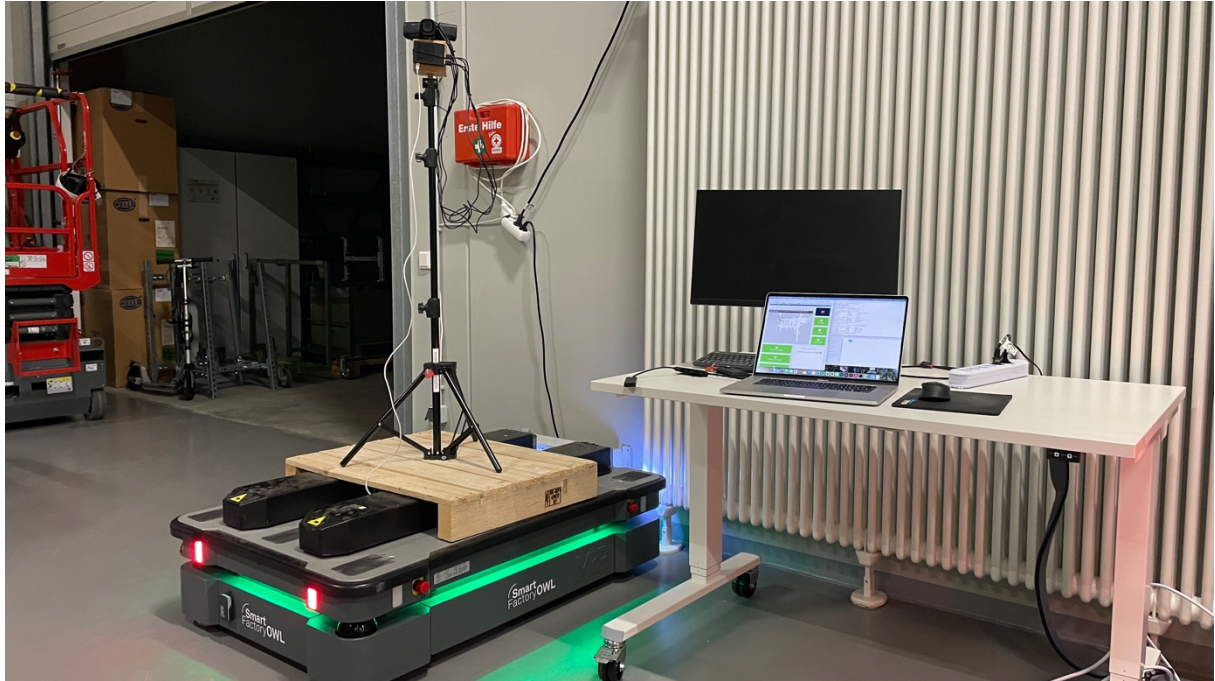


Abbildung 4: Überblick über das gesamte System bestehend aus MiR, Kamera Setup sowie Macbook für den Fernzugriff [10]

5.4 Vergleichskriterien für Evaluierung

Eine detaillierte Evaluierung der Punktwolkenqualität erfordert quantitative Metriken, die sowohl die geometrische Genauigkeit als auch die visuelle Qualität bewerten. Die zwei am häufigsten verwendeten Qualitätsmetriken in der Punktwolkenanalyse sind der Peak Signal-to-Noise Ratio (PSNR) und der Structural Similarity Index (SSIM).

Laut einer Studie von Zhang et al. (2022) zur Punktwolkengenerierung mittels Photogrammetrie und Gaussian Splatting liegt der durchschnittliche PSNR-Wert für Gaussian Splatting bei etwa 35 dB, während klassische Photogrammetrie-Methoden Werte zwischen 30–33 dB erreichen. Höhere PSNR-Werte deuten auf eine bessere Erhaltung der Bilddetails und eine geringere Verzerrung hin. Der SSIM-Wert von Gaussian Splatting wurde in Experimenten mit 0.92 bewertet, was auf eine hohe strukturelle Ähnlichkeit mit der Referenzszene hinweist, während Photogrammetrie nur Werte um 0.85 erreichte [40].

Zusätzlich ist die Punktdichte ein entscheidender Faktor. Gaussian Splatting erreicht eine Punktdichte von ca. 90 % im Vergleich zu LiDAR-Systemen, während klassische Photogrammetrie in der Regel nur 75–80 % erzielt (9).

6 Mögliche Anwendungsfälle

Das entwickelte 3D-Rekonstruktionssystem eröffnet Möglichkeiten innerhalb und außerhalb von industriellen Smart Factories; Allein der kreative Einfallsreichtum der Anwender setzt die Grenzen. Über die Produktionsebene hinaus drängen sich innovative Anwendungen auf, die noch vor Jahren undenkbar schienen. In großen wie kleinen Unternehmen wie auch Organisationen erleichtert die digitale Erfassung von räumlicher Information alle Abläufe, wobei sich ständig neue Geschäftsfelder auftun – der Wettbewerbsvorteil liegt auf der Hand.

6.1 Digitale Zwillinge für Produktionsumgebungen

Ein faszinierender Einsatzbereich zeigt sich bei der Entwicklung digitaler Zwillinge, die in Fabriken zum Einsatz kommen. Digitale Zwillinge erschaffen virtuelle Abbildungen von Produktionsanlagen, wodurch sich Prozesse sowie Zustände live verfolgen lassen – sie spiegeln damit die physische Realität in der digitalen Dimension wider. In der SmartFactoryOWL nutzt man durchgängig vernetzte Computermodelle; Sie überwachen nicht nur den laufenden Betrieb, sondern ermöglichen zudem eine vorausschauende Instandhaltung der Maschinen. Eine solche Herangehensweise optimiert die Organisation der Abläufe und minimiert unerwartete Stillstände in der Produktion; Davon profitieren besonders Betriebe mit kleiner und mittlerer Größe.

6.2 Automatisierte Inventarisierung im Einzelhandel

Bereits umsetzbare Anwendungen könnten folgende sein: Das Scannen von kleinen, aber insbesondere großen Produkten für Endkunden bietet vielseitige Anwendungsmöglichkeiten. Beispielsweise können große, teure Bäume im Gartencenter, Gewächshäuser oder individuell angefertigte Produkte schnell digital erfasst und per AR direkt auf dem Grundstück des Kunden visualisiert werden. Dadurch lassen sich Fehlkäufe und Retouren vermeiden, während Betriebe ihre Produkte leichter in Onlineshops integrieren können. Gleichzeitig erhält der Endkunde eine bessere Vorstellung davon, wie ein bestimmtes Produkt, etwa ein kostspieliger Olivenhain, in seiner eigenen Umgebung – dem Garten – wirken würde.

Auch im Einzelhandel bietet die Technologie Vorteile: Märkte können effizient gescannt werden, um Aufsteller oder größere Werbeflächen optimal zu platzieren. Besonders Supermärkte, die regelmäßig sperrige Pappaufsteller oder Produktwerbung erhalten, profitieren davon. Durch die digitale Erfassung des Marktes und der Werbematerialien kann eine präzisere Planung erfolgen, wodurch Zeit gespart, Platz effizient genutzt und die Präsentation sowohl für Kunden als auch für Werbetreibende attraktiver gestaltet wird.

Ein faszinierender Bereich eröffnet sich im Einzelhandel – besonders Gartencenter sowie Supermärkte profitieren von den innovativen Möglichkeiten. Über präzise 3D-Scans entsteht ein digitales Abbild aller Produkte, die sich in den Regalen befinden. Die automatische Erfassung löst mühsame Handarbeit ab und vermeidet Irrtümer, die sonst beim Zählen entstehen. In den vernetzten Systemen übernehmen selbstlernende Programme die Sortierung der Artikel, deren

Stückzahlen sofort gespeichert werden; Daraus ergeben sich laufend neue Wege zur Verbesserung der Lagerverwaltung.

6.3 Virtuelle Museen und Denkmalschutz

Museen als auch Denkmäler bewahren durch digitale Dokumentation ihre wertvollen Schätze. Ein System, das sich durch seine Präzision auszeichnet, erstellt dreidimensionale Abbildungen – hierdurch entstehen detailgetreue Aufzeichnungen von Exponaten als auch von Ausstellungsräumen. Das Hermannsdenkmal in Detmold und das Freilichtmuseum OWL öffnen dank dieser Technik bald ihre Pforten für Interessierte, die weit entfernt leben oder sich nicht vor Ort bewegen können. Die digitalen Speicherungen ermöglichen einen universellen Zugriff auf kulturelle Schätze; virtuelle Besichtigungen wie auch Mitmach-Ausstellungen eröffnen völlig neuartige Perspektiven auf historische Zeugnisse [51].

6.4 Einsatz im Bauwesen und Architektur

Solche Scanner sind zwar bereits im Einsatz aber werden meist nur von großen, umsatzstarken Unternehmen benutzt und sind sehr kostspielig in der Anschaffung. In der Baubranche revolutioniert das 3D-Scanning die Gebäudedokumentation; Ingenieure erfassen dadurch Bauwerke, die sich bis ins kleinste Detail vermessen lassen. Die Technologie schafft eine Basis für gründliche Instandsetzungen wie auch Modernisierungen. Das virtuelle Abbild hilft bei der Simulation von Umbauten – die Entwicklung der Baustelle lässt sich lückenlos nachverfolgen. Unter wiederholter Anwendung dieser Methode entstehen Aufzeichnungen, aus denen sich strukturelle Schwachstellen bereits im Entstehen erkennen lassen [49], [50].

6.5 Erweiterte Anwendungen in der Industrie 4.0

Das System bietet neben den vorgestellten Anwendungen auch großes Potenzial für Virtual Production und Filmsets. Besonders in der Filmindustrie wird zunehmend auf 3D-Scanning-Technologien gesetzt, um digitale Umgebungen effizient zu rekonstruieren. Gaussian Splatting könnte hier als kostengünstige Alternative zu hochpreisigen LiDAR-Scannern eingesetzt werden, um Filmsets virtuell zu erfassen und fotorealistische Hintergründe in Echtzeit zu rendern. Besonders für VR und AR Anwendungen könnte das System eine neue Möglichkeit schaffen, große Umgebungen mit hoher Detailgenauigkeit zu erfassen und für interaktive Medien nutzbar zu machen.

Neben der Raumüberwachung als auch Dokumentation innerhalb von Betriebsgebäuden – wie in der SmartFactoryOWL – erschließen sich fesselnde neue Einsatzgebiete. Eine Technologie, die über industrielle Grenzen hinauswächst und in Supermärkten, Museen, auf Baustellen, in Rettungswachen und sogar in Gartenzentren Einsatz findet. Bahnbrechende Möglichkeiten zur 3D-Digitalisierung und -Rekonstruktion von Räumen, die sich ständig verändern, oder einzelnen Produkten entstehen; Dadurch lassen sich virtuelle Abbilder erschaffen. Die Vorteile dieser Innovation nutzen nicht nur Großkonzerne, sondern auch kleine Geschäfte, die mit preiswerten

Kamerasystemen ihre Räume dreidimensional erfassen. Eine digitale Umwälzung verändert alles von Grund auf.

7 Praktische Umsetzung und Herausforderungen

Während der Implementierung des Projekts traten unerwartete Herausforderungen auf, die insbesondere die Kommunikation mit der API des Roboters und die Qualität der aufgenommenen Bilder betrafen.

7.1 FTP Server und Fernzugriff

Um einen einfachen Zugriff zu ermöglichen und Tests schneller durchführen zu können, musste erst eine geeignete Arbeitsumgebung geschaffen werden. Denn der Aufbau via Monitor und HDMI Kabel und weiteren Peripherie Geräten per Kabel war nicht ideal. Es wurden mehrere Ansätze iteriert, zum Beispiel ein Remote Zugriff vom Macbook und Windows PC auf den Raspberry Pi über Teamviewer. Allerdings zieht dies Leistung des ohnehin nicht sonderlich starken Raspberry Pis. Deshalb findet der Fernzugriff per SSH Verschlüsselung im gleichen Netzwerk statt. So fehlt zwar das grafische Interface, aber alle Anpassungen können auch über die Kommandozeile vorgenommen werden und die konstante Leistung rechtfertigt diesen Nachteil.

Um die aufgenommenen Bilder automatisch auf dem Pi zu speichern und per FTP abzurufen, wurde ein Shell-Skript (Abbildung 24) implementiert. Dieses speichert die Bilder standardmäßig in `/home/sam/FTP/files`. Erste Schwierigkeiten ergaben sich durch fehlende Schreibrechte, die durch das Anlegen der Verzeichnisse mittels `mkdir -p` behoben wurden. Zudem führten Firewall-Einstellungen und Konfigurationsprobleme des FTP-Servers dazu, dass der Zugriff zunächst nicht möglich war. Eine entsprechende Anpassung der `ufw`-Regeln und der `vsftpd`-Konfiguration löste das Problem

7.2 Erweiterung des MIR für eine USB-C Schnittstelle

Die Erweiterung des MiR-Roboters um eine USB-C-Schnittstelle ist eine essenzielle Voraussetzung für den Betrieb des gesamten Systems, da sie eine zuverlässige Datenübertragung und Stromversorgung externer Geräte ermöglicht. Ohne diese Schnittstelle wäre das System ausschließlich über Akkulösungen realisierbar, was nicht nur die Betriebsdauer einschränken, sondern auch zusätzliche Wartungs- und Ladeprozesse erfordern würde – ein klarer Nachteil in einer automatisierten Umgebung. Die USB-C-Schnittstelle gewährleistet eine stabile Verbindung für hochauflösende Kameras und ermöglicht eine gleichzeitige Datenübertragung und Energieversorgung, wodurch eine kontinuierliche Echtzeitverarbeitung der Bilddaten ohne Unterbrechungen sichergestellt wird.

Für die Integration der Schnittstelle musste die Bodenplatte des MiR-Roboters abgeschraubt und eine Öffnung in das Metallgehäuse gesägt werden, um die benötigte Hardware sicher zu verbauen. Zudem war die direkte Nutzung der vorhandenen 24V-Stromversorgung des Roboters nicht möglich, weshalb ein Spannungswandler installiert wurde, der die 24V auf 5V konvertiert, um eine kompatible und stabile Stromversorgung für die angeschlossenen USB-

C-Geräte zu gewährleisten. Diese Anpassung stellt einen kritischen Punkt dar, um das Gesamtsystem effizient und autonom betreiben zu können.

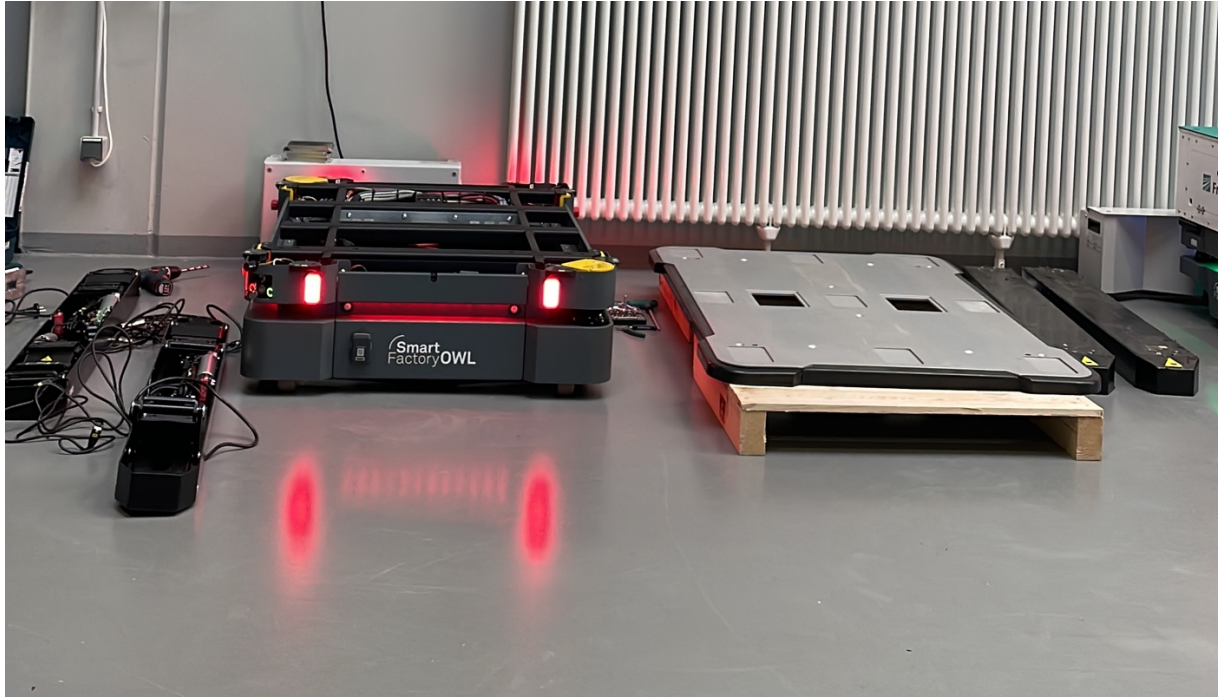


Abbildung 5: MiR mit abgeschraubter Bodenplatte sowie Hydrauliksystem für die Errichtung einer USB-C Schnittstelle [10]

7.3 Entwicklung der Missionsplanung für den Roboter

7.3.1 Erster Ansatz: Steuerung über das Web-Interface des MiR

Zunächst wurde eine Missionsplanung vollständig über das **Web-Interface des MiR-Roboters** getestet. Dabei wurden **Wegpunkte in festen Intervallen (50 cm / 1 m)** manuell im Interface eingetragen und als einzelne Mission abgearbeitet. Dies erwies sich als **extrem zeitaufwendig und unflexibel**, da jeder Punkt manuell editiert und angepasst werden musste.

Ein großes Problem entstand, als **die Karte des MiR-Roboters mehrfach zurückgesetzt oder neu erstellt wurde** – sei es durch Software-Updates oder durch eine erneute Kartenerfassung der Umgebung. Dies führte dazu, dass die gesamte Missionsplanung **erneut durchgeführt werden musste**, da die Wegpunkte nicht mehr mit der neuen Karte übereinstimmten.

Zusätzlich kam es zu **asynchronen Kameraauslösungen**, da der Roboter die Wegpunkte mit einer variablen Geschwindigkeit abfuhr, während das Kamera-Skript **nicht exakt mit der Bewegung des Roboters synchronisiert** war. Die Folge waren unscharfe oder falsch getimte Bilder, da die Kameras während der Fahrt ausgelöst wurden

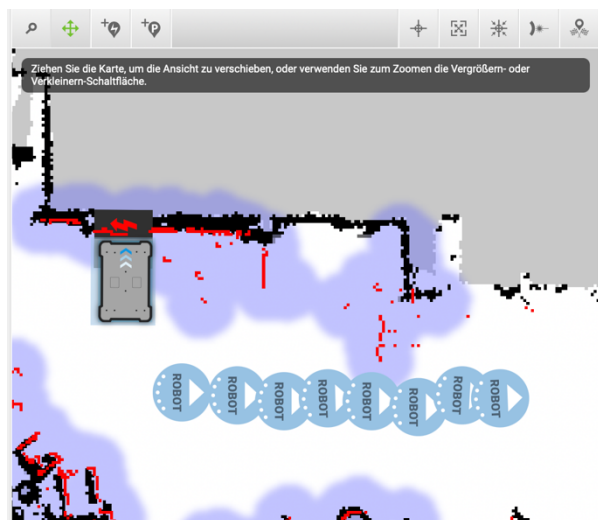


Abbildung 7: Grafisches Web-Interface des MiR und die erstellten Wegpunkte im ca. 1m Abstand [10]

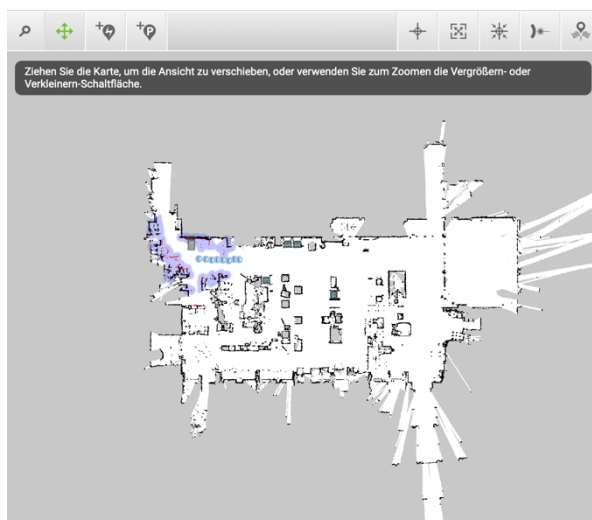


Abbildung 6: Grafisches Web-Interface des MiR im gesamten Kartenausschnitt [10]

7.3.2 Zweiter Ansatz: Steuerung über API mit automatisierter Wegpunkt-Erstellung

Um mehr Flexibilität zu erreichen, wurde ein zweiter Ansatz erprobt, bei dem der Roboter über die API-Schnittstelle direkt gesteuert wurde. Die Idee war, dass der Roboter automatisch alle 50 cm / 1 m anhält, um eine präzise Aufnahme der Umgebung zu gewährleisten. Dieser Ansatz brachte neue Herausforderungen mit sich.

Fehlgeschlagene API-Verbindungen: Erste Implementierungen des Skripts enthielten Fehler in der Authentifizierung, wodurch keine zuverlässige Kommunikation mit der API möglich war.

Ungenauigkeiten durch Rotation: Da sich der Roboter an strategischen Punkten rotieren muss, ergaben sich Abweichungen in den Endpositionen. Bereits kleine Fehler in der Positionierung führten zu größeren Abweichungen, sodass die geplanten Rotationen nicht exakt durchgeführt wurden.

Anstatt eine einzelne Mission mit mehreren Wegpunkten anzulegen, wurde ein anderer Workflow etabliert: Der Roboter steuert jeden Punkt als separate "Gehe zu Position"-Mission an, die dann wie eine durchgehende Mission wirkt. Dadurch wurde das System robuster gegenüber Kartenänderungen und unvorhergesehenen Hindernissen.

7.3.3 Finaler Ansatz: Strategische Punkte und dynamische Wegpunkte

Die finale Version der Missionsplanung in der SmartFactory berücksichtigt strategisch festgelegte Wegpunkte, an denen sich der Roboter sicher dreht und stoppt, um eine präzisere Aufnahme zu ermöglichen. Die Strategie funktioniert folgendermaßen:

Der MiR berechnet die Distanz zwischen zwei Wegpunkten in einem kartografischen Koordinatensystem. Diese Distanz wird in Intervalle von 50 cm / 1 m unterteilt, wodurch automatisch Stoppunkte generiert werden. Nach jeder festgelegten Distanz wird die Kamera ausgelöst, sobald die Geschwindigkeit des Roboters auf Null fällt. Falls der Roboter eine Rotation durchführen muss, stoppt er an vordefinierten Drehpunkten und fährt erst nach Abschluss der Rotation weiter.

Diese Methode führt zu einer präzisen und synchronisierten Erfassung der Umgebung, ohne dass Hindernisse das Aufnahmeverfahren beeinflussen. Allerdings hat diese Strategie zur Folge, dass die Gesamtdauer einer Durchfahrt steigt, da der Roboter durch das häufige Stoppen und Beschleunigen mehr Zeit benötigt.

7.4 Synchronisation der Kameras und Bildaufnahme

Die größte Herausforderung bei der synchronisierten Bildaufnahme war die unterschiedliche Auslösezeit der USB-Kameras, da das Betriebssystem die Gerätepriorisierung individuell vornimmt. Dies führte zu minimalen Zeitabweichungen zwischen den Bildern verschiedener Kameras. Zur Lösung dieses Problems wurden folgende Maßnahmen ergriffen:

Manuelle Gerätezuweisung der USB-Kameras: Da Linux jedem angeschlossenen Gerät eine dynamische ID zuweist, wurde ein Skript entwickelt, das die Kameras persistent einer festen ID zuordnet.

Synchronisierte Trigger-Mechanismen: Das Skript zur Kameraauslösung wurde so angepasst, dass die Aufnahme erst nach einer Abfrage der Roboter-Geschwindigkeit **erfolgt**. Dies stellt sicher, dass die Bilder nur dann aufgenommen werden, wenn der Roboter stillsteht, wodurch Bewegungsunschärfe vermieden wird.

Zukünftige Optimierungen könnten den Einsatz von synchronisierten Hardware-Trigger oder spezialisierten Kamerasystemen mit exakten Bildauslösungen umfassen.

7.4 Probleme mit API-Kommunikation und Hotfixes

Ein wesentliches Problem bestand in der fehlerhaften Abfrage des Roboterstatus über die API. Das Skript musste als Einzeiler ohne Sonderzeichen formuliert werden, was die Fehlersuche erheblich erschwerte. Dies führte zu Verzögerungen in der Entwicklung, da die Bewegung des Roboters erst korrekt überwacht werden musste, bevor die Kameras synchronisiert ausgelöst werden konnten.

Die Integration von ChatGPT als unterstützendes Tool für die Codegenerierung hat sich in vielen Bereichen als hilfreich erwiesen, zeigte jedoch Schwächen bei der API-Anbindung des Roboters. Hier führte die fehlerhafte Erstellung von Einzeilern für den Authentifizierungsschlüssel zu erheblichen Verzögerungen. Zusätzlich führte ein Update des MIR-Roboters dazu, dass Benutzerrechte zurückgesetzt und Wegpunkte neu angelegt werden mussten, was den Workflow erheblich störte.

7.5 Bildqualität und USB-Webcam Probleme

Ein weiteres Problem war die Bildqualität der USB-Kameras. Während der Livestream über den Raspberry Pi und das MacBook eine klare, rauschfreie Ausgabe lieferte, waren die aufgenommenen Bilder oft verrauscht und leicht verschwommen. Bei den Tests wurde klar festgestellt, dass Fokus, Belichtung und Weißabgleich zufällig verändert wurden. Die Ursache ist, dass die Kameraeinstellungen bei jeder neuen Aufnahme überschrieben wurden. Alle Hotfixes in Python konnten das Problem nicht lösen.

7.5.1 Lösungsansätze zur Verbesserung der Bildqualität

Ein kritischer Faktor, der die Entwicklung des Projekts beeinflusste, war die Firmware-Kompatibilität der eingesetzten USB-Kameras. Während erste Tests mit Anker PowerConf Kameras durchgeführt wurden, stellte sich heraus, dass deren interne Firmware automatisch Parameter wie Belichtung und Weißabgleich überschreibt, wodurch eine kontinuierliche Bildqualität nicht sichergestellt werden konnte.

Dieser Umstand führte dazu, dass die Kameras gegen Logitech Brios ausgetauscht werden mussten, da diese eine vollständige manuelle Steuerung der Parameter ermöglichten. Allerdings waren die Logitech-Kameras drei Mal so teuer, was das Budget für das Projekt erheblich belastete. Zusätzlich mussten die für die Tests verwendeten Kameras nach dem Experiment zurückgegeben werden, wodurch keine weiteren Testfahrten mehr möglich waren.

Zukünftige Arbeiten sollten daher eine Liste kompatibler Kameras enthalten, um ähnliche Probleme in anderen Projekten zu vermeiden. Zusätzlich könnte ein Firmware-Check-Tool in das Skript integriert werden, das vor der Bildaufnahme prüft, ob sich die Kameraeinstellungen automatisch zurücksetzen.

Eine weitere Optimierung wäre, die gesamte Kamera-Software direkt auf den Raspberry Pi zu portieren, um kompatiblere Alternativen zur USB-Webcam-Nutzung zu evaluieren.

Zu Beginn waren die Bilder verrauscht und unscharf, insbesondere bei weiter entfernten Objekten. Obwohl die mittels `v4l2-ctl` gesetzten Fokuswerte gespeichert wurden, wurden sie nach einem Neustart zurückgesetzt. Dies wurde durch ein `set_camera.sh`-Skript versucht zu beheben, das die gewünschten Kameraeinstellungen nach jedem Neustart erneut setzt. Zudem wurde der UVC-Treiber vor jeder Bildaufnahme entladen und neu geladen (`modprobe -r uvcvideo && modprobe uvcvideo`). Durch eine Kombination der Kameraeinstellungen und des `fswebcam`-Befehls in einer einzigen Zeile konnte die Bildqualität trotzdem nicht verbessert werden.

Auf Anfrage des Herstellers Anker wurde verifiziert, dass die interne Firmware diese Einstellungen überschreibt. Deshalb sind diese Kameras nicht für das System geeignet und mussten zurückgeschickt werden.

7.6 Camera Alignments

Es wurden verschiedene Experimente mit den Parametern `auto_exposure`, `exposure_time_absolute` und `focus_absolute` durchgeführt. Längere Belichtungszeiten verbesserten dunkle Bereiche, erhöhten jedoch das Rauschen. Die optimale Balance wurde bei `exposure_time_absolute=1000`, `focus_absolute=400` und `auto_exposure=1` gefunden. Trotz aller Anpassungen lieferte die AnkerWork-Software auf dem Mac eine erheblich bessere Bildqualität als `v4l2-ctl` auf dem Raspberry Pi. Dies deutet darauf hin, dass Treiber in der Kamera (in Verbindung mit dem Raspberry Pi) bestimmte Einstellungen überschreiben oder nicht optimal mit der Hardware interagieren.

Spiegelungen als auch Floater erschwerten die Ausrichtung der Kameras - jene Störfaktoren entstehen, falls der Roboter zu große Abstände zwischen den Wegpunkten einplant oder die Überlappung der Bilder nicht ausreicht. Eine Verdichtung der Aufnahmepositionen oder der Umbau von sechs auf acht Kameras würde die Überlappung maximieren; dadurch entstünden bessere Punktwolken. Der Roboter könnte alternativ zusätzliche Runden in der SmartFactory absolvieren und mehr Datenmaterial erfassen. Die Berechnung der Splats beansprucht laut Tests mindestens 30 Minuten; der Zeitaufwand variiert dabei je nach k-Splat-Anzahl und eingesetzter Hardware [9].

7.7 Sensoren und Protective Fields

Ein weiteres Problem bei der Steuerung des MiR-Roboters war die Automatik der Sicherheitszonen (Protective Fields). Diese sensor- und softwarebasierten Bereiche um den Roboter führen dazu, dass er unnötig stoppt oder abrupte Kurskorrekturen vornimmt, insbesondere in engen Gängen oder bei unerwartetem Fußgängerverkehr in der Halle. Eigentlich in Produktionsumgebungen von Vorteil um Unfällen vorzubeugen, aber für das Experiment unvorteilhaft.

Frühere Tests zeigten, dass der Roboter in einigen Abschnitten unnötig abbremste oder zu vorsichtige Navigationsentscheidungen traf, was die Gesamtdauer einer Durchfahrt erheblich verlängerte. Durch eine Anpassung der dynamischen Sicherheitsfelder konnte das Manövrieren effizienter gestaltet werden. Beispielsweise wurde die Sensibilität in Bereichen mit wenig Hindernissen reduziert, während kritische Abschnitte mit einer hohen Dichte an Maschinen oder Personen weiterhin strenge Sicherheitszonen behielten. Ebenso leitete der Roboter unerwartete Stopps ein und setzte teilweise Missionen fehlerhaft fort oder übersprang Wegpunkte.



Abbildung 8: MiR und rotem LED-Indikator für einen blockierten Sensor im Frontbereich [10]

7.8 Optimierungsmöglichkeiten durch maschinelles Lernen

Eine übersichtliche Bedienoberfläche und die Verbindung zur Cloud vereinfachen die Handhabung der Programme. Über einen künftigen Hochleistungsrechner am KreativInstitut OWL lassen sich Kalkulationen per Mausklick anstoßen; Automatische E-Mails informieren anschließend über die Resultate. Eine Software, die sich selbst weiterentwickelt, ermittelt automatisiert die besten Parameter – dadurch beschleunigen sich die Arbeitsabläufe. Leistungsstarke Netzwerke wie DnCNN filtern störende Pixelfehler heraus; Gleichzeitig bestimmen lernfähige (Machine-Learning-)Systeme durch Deep Reinforcement die Standorte, von denen aus Kameras ihre Umgebung optimal erfassen [52].

9 Diskussion und Ausblick

9.1 Interpretation der Ergebnisse

Die nachfolgenden Bilder behandeln die einzelnen erstellten visualisierten Punktwolken und wie diese für deren Verwendungszweck einzuordnen sind. Aufgrund der in Kapitel 7 und insbesondere 7.5 unvorhergesehenen Herausforderungen und nicht vollständig erarbeiteten Lösungsansätze, spiegeln diese Modelle nur einen Zwischenstand der Arbeit wider. Es wird weiter an möglichen multisensorischen Kamera Setups und einer Verbesserung des gesamten Workflows gearbeitet.

Die bisherigen Tests haben gezeigt, dass das entwickelte 3D-Rekonstruktionssystem prinzipiell funktioniert, jedoch noch einige Herausforderungen bestehen. Besonders die manuell rekonstruierten Testläufe mit einer DSLR-Kamera, bei denen acht Bilder aufgenommen wurden, bestätigen die grundsätzliche Funktionsfähigkeit des Systems. Die mit Gaussian Splatting generierten Punktwolken zeigen, dass die Methode in der Lage ist, 3D-Modelle zu erzeugen. Dennoch sind noch einige technische Hürden zu bewältigen, bevor das System voll einsatzfähig ist. Ein zentrales Problem ist derzeit die Ausrichtung der Kameras (Camera Alignment) und die Überlappung der Bilder. Die bisher erfassten Bilder weisen teilweise zu geringe Überlappungen auf, was zu Artefakten in den Punktwolken führt. Eine exakte Kamerakalibrierung und eine bessere Synchronisation zwischen den einzelnen Aufnahmen sind essenziell, um die Qualität der 3D-Rekonstruktion zu verbessern. Eine Möglichkeit zur Optimierung könnte sein, die Positionen der Kameras am Roboter so anzupassen, dass sich größere Überlappungen zwischen den Bildern ergeben. Alternativ könnten zusätzliche Kameras integriert oder mehrere Durchläufe mit leicht versetzten Positionen durchgeführt werden, um mehr Bildmaterial zu gewinnen. Neben der Bildausrichtung stellt auch die Bildqualität der USB-Webcams eine Herausforderung dar. Während die DSLR-Kamera in den Tests eine hohe Detailtreue bot, leiden die Webcams unter Problemen mit Fokus, Belichtung und Weißabgleich. Insbesondere der automatische Reset der Kameraeinstellungen erschwert eine konsistente Bildaufnahme. Dieses Problem könnte durch alternative Kameramodelle mit einer besseren Software-Steuerung oder durch den Einsatz externer Kalibrierungstools gelöst werden.

Die Software lenkt den Roboter gut, zeigt aber Schwächen. Er navigiert zwischen festgelegten Punkten; manchmal steht er dabei nicht exakt an der richtigen Stelle. Seine Kamera nimmt dann Bilder aus falschen Winkeln auf - das verschlechtert die Überlappung der Aufnahmen und reduziert die Qualität beim Erstellen von 3D-Modellen. Ein neues Programm, das sich automatisch an Hindernisse und Gegebenheiten im Raum anpasst, würde besser funktionieren. Die Verwendung eines flexibleren Navigationsalgorithmus, der sich dynamisch an die Umgebung anpasst, könnte hier eine Verbesserung bringen.

Trotz dieser Herausforderungen zeigt die Arbeit, dass Gaussian Splatting als kosteneffiziente Alternative zu bestehenden 3D-Rekonstruktionsverfahren großes Potenzial besitzt. Die erzeugten Punktwolken sind speichereffizient und lassen sich schneller berechnen als herkömmliche

Photogrammetrie-Modelle. Besonders für den industriellen Einsatz, etwa in der SmartFactory-OWL, bietet diese Methode eine skalierbare Lösung zur Erstellung digitaler Zwillinge.



Abbildung 10: Rendering aus Postshot zur Visualisierung des Gaussian Splat Models bei Verwendung von vier Logitech Brio Kameras auf dem MiR [10]



Abbildung 9: Rendering aus Postshot zur Visualisierung des Gaussian Splat Models bei Verwendung von vier Logitech Brio Kameras auf dem MiR – Das Gesamtmodell ist aufgrund der Camera Alignments nicht brauchbar [10]



Abbildung 11: Rendering aus Postshot zur Visualisierung des Gaussian Splat Models bei Verwendung von vier Logitech Brio Kameras auf dem MiR [10]



Abbildung 12: Rendering aus Postshot zur Visualisierung des Gaussian Splat Models bei Verwendung von vier Logitech Brio Kameras auf dem MiR [10]



Abbildung 13: Rendering aus Postshot zur Visualisierung des Gaussian Splat Models bei Verwendung einer iPhone 12 Pro Kamera [10]



Abbildung 14: Rendering aus Postshot zur Visualisierung des Gaussian Splat Models bei Verwendung einer iPhone 12 Pro Kamera [10]



Abbildung 15: Rendering aus Postshot zur Visualisierung des Gaussian Splat Models bei Verwendung einer Nikon D7500 DSLR bei einer Rekonstruktion der MiR Mission durch einen Menschen [10]

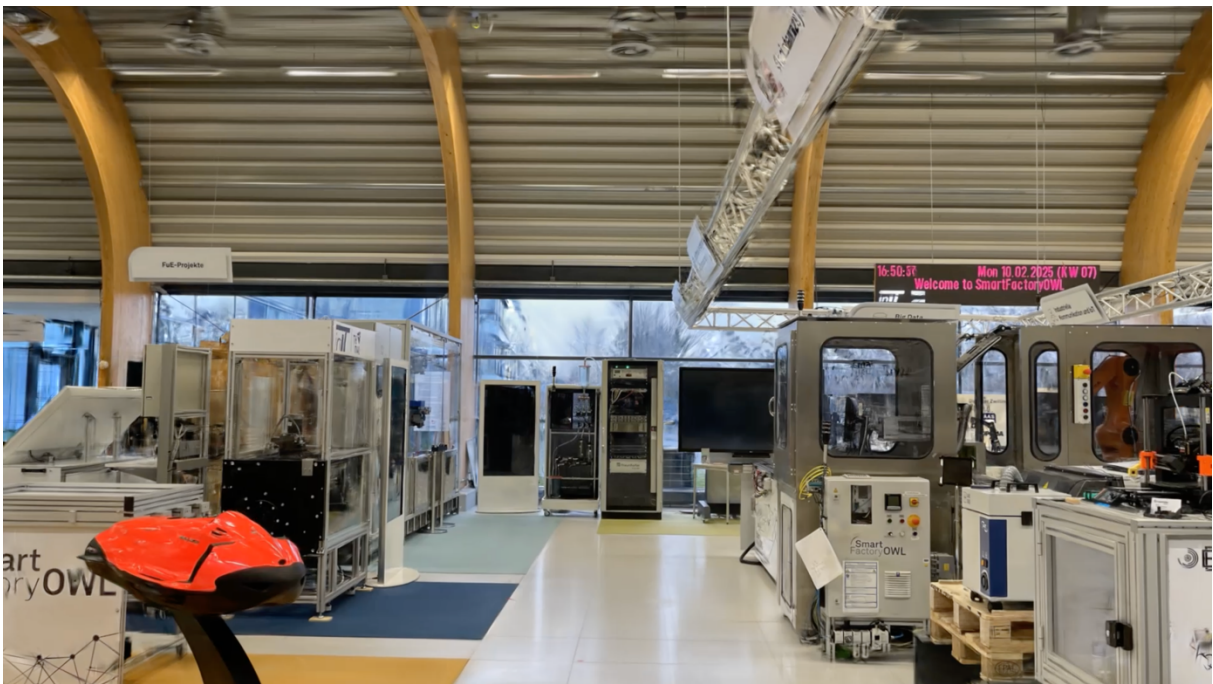


Abbildung 16: Rendering aus Postshot zur Visualisierung des Gaussian Splat Models bei Verwendung einer Nikon D7500 DSLR bei einer Rekonstruktion der MiR Mission durch einen Menschen [10]



Abbildung 17: Rendering aus Postshot zur Visualisierung des Gaussian Splat Models bei Verwendung einer Nikon D7500 DSLR bei einer Rekonstruktion der MiR Mission durch einen Menschen [10]



Abbildung 18: Punktwolke aus Postshot zur Visualisierung des Gaussian Splat Models bei Verwendung einer Nikon D7500 DSLR bei einer Rekonstruktion der MiR Mission durch einen Menschen [10]

9.1.1 Fotos mit automatischem und manuellem (festem) Weißabgleich

Da nur die Logitech Brio ein zuverlässiges Fotogenerierung ermöglichte, beschränkt sich dieser Vergleich auf nur ebendieses Kamerasetup. Es stellte sich heraus, dass ein manueller Wert (z.B. 4000K) sich verändernde Ergebnisse erzielte. Eigentlich sollte bei einem festen Wert sich der Weißabgleich nicht ändern bzw. ähnliche Resultate erzielen. Die ausgegebenen Fotos kommen aber zu einem anderen Schluss. Bei einem automatischen Weißabgleich der Kamera haben die Fotos ähnlichere Farbwerte und führen so auch zu besseren Splats. Die mit dem festen Weißabgleich generierten Bilder passen vom Farbumfang kaum zu einander. Folgende vier Bilder zeigen einen festen Weißabgleich und die anschließenden vier dieselben Bildausschnitte aber mit einem automatischen Weißabgleich der Logitech Brio Webcams.



Abbildung 20: Kamera 1 der vier Logitech Brio Kameras und festem Weißabgleich [10]



Abbildung 19: Kamera 2 der vier Logitech Brio Kameras und festem Weißabgleich [10]



Abbildung 21: Kamera 3 der vier Logitech Brio Kameras und festem Weißabgleich [10]



Abbildung 22: Kamera 4 der vier Logitech Brio Kameras und festem Weißabgleich [10]

9.2.1 Kosten Vergleich mit anderen Systemen

Tabelle 2: Kosten Vergleich von den evaluierten Kamerasystem auf dem MiR [10]

Komponente	Kosten					
8x USB-Kameras (Anker C200)	480€ (8 x 60€)					
4x USB-Kamera (Logitech Brio)		600€ - 800€ (4 x 150 - 200€)				
8x USB-Kamera (Logitech Brio)			1200€ - 1600€ (8 x 150€ - 200€)			
4x Raspberry Pi Kamera und 4x Objektive				240€ (4 x 60€)		
4 x DSLR Kameras (Nikon D7500)					4000€ (4 x ~1000€)	
4 x iPhone 12 Pro						2400€ (4 x ~800€)
Roboterplattform (MiR 600)	~30.000€					
Raspberry Pi 4 (4GB)	60€					
NVIDIA RTX 4070 GPU	700€					
CPU (Ryzen 9 5950X)	500€					
RAM (64GB DDR4)	250€					
Sonstige Hardware (Kabel,	75€					

<i>Netzteile, Stativ)</i>						
Gesamt- kosten (ohne MiR)	2.065€	2.185€ - 2.385€	2.785 – 3.185€	1825€	5.585€	3.985€

Die Gesamtkosten sind nur aufgrund des geschätzten Preises des MiR höher, welcher mit Abstand die teuerste Komponente ist. Dieser ist für den Versuchsaufbau in der SmartFactoryOWL unerlässlich und wird auch bei anderen Systemen benutzt und wird deshalb nicht in der Preisübersicht berücksichtigt. Wenn die Logitech Brio Kameras aus dem Inventar der SmartFactoryOWL stammen, kann dies gegengerechnet werden und so wird die Kalkulation angepasst.

Die Wahl der Kamera beeinflusst die Gesamtkosten erheblich:

Anker C200 (8x): Die günstigste Option (2.065 €) mit akzeptabler Bildqualität, jedoch limitierten Einstellungsmöglichkeiten.

Logitech Brio (4x oder 8x): Liefert höhere Bildqualität, aber die Kosten steigen auf 2.185–3.185 €.

Raspberry Pi Kamera (4x): Sehr günstig (1.825 €), aber möglicherweise schlechtere Qualität.

DSLR-Kameras (4x Nikon D7500): Höchste Bildqualität, aber auch die teuerste Option (5.585 €).

iPhone 12 Pro (4x): Gute Qualität, aber fraglich, ob es für industrielle Anwendungen praktikabel ist, zumal die Datenverarbeitung schwierig ist.

9.2.2 Vergleich mit kommerziellen 3D-Scannern

Tabelle 3: Kosten Vergleich mit einem evaluierten Kamerasystem zu anderen kommerziellen 3D-Scannern [10]

System	Kosten (EUR)	Vorteile	Nachteile
Entwickeltes System (4x Logitech Brio)	2.185€ - 2.385€	Günstigste Lösung, hohe Flexibilität	Weniger präzise als Li- DAR
Matterport Pro2	~5.000 €	Einfache Bedienung, Cloud-Dienste	Weniger flexibel, be- grenzte Nutzung
FARO Focus Premium	~15.000 €	Sehr hohe Präzision	Teuer, aufwendige Ka- librierung
Leica BLK360	~20.000 €	Kompakt, sehr genau	Hohe Kosten, spezielle Software nötig

Im Vergleich zu anderen Systemen bietet das entwickelte Setup zu kommerziellen Alternativen eine deutlich kostengünstigere Alternative. Insbesondere für KMU oder flexible Anwendungen

ist dies eine sinnvolle Alternative. Die Anschaffung von hochwertigen Kameras wie DSLR oder iPhones würde die Kosten in den Bereich kommerzieller Scanner treiben, ohne unbedingt deren Präzision zu erreichen. Die günstigste Alternative bleibt das Raspberry Pi System, welches sich aber in den Versuchen als nicht praktikabel herausgestellt hat.

9.3 Beantwortung der Forschungsfragen

Eine der zentralen Forschungsfragen dieser Arbeit lautet:

„Wie kann ein mobiles, kosteneffizientes Kamera-Setup für die präzise und speichereffiziente 3D-Rekonstruktion einer dynamischen Produktionsumgebung entwickelt und optimiert werden?“

Die Tests beweisen, wie ein System mit USB-Kameras und Gaussian Splatting funktionieren kann; die Aufbauten der Kameras ließen sich gut vergleichen. Logitech Brio Webcams zusammen mit DSLR-Kameras erzeugten brauchbare Resultate, wenn auch mit Verbesserungspotenzial. Aber DSLR-Geräte überzeugten durch scharfe Details und als vorläufiges Ergebnis.

Ein weiterer wichtiger Aspekt war die Frage, wie sich das entwickelte System im Vergleich zu bestehenden Marktlösungen wie Photogrammetrie oder LiDAR-Systemen schlägt. Die Tests ergaben, dass Gaussian Splatting zwar nicht die höchste geometrische Präzision erreicht, jedoch in Bezug auf Rechenzeit und Speicherverbrauch deutliche Vorteile bietet. Besonders für Anwendungen, bei denen eine schnelle und flexible 3D-Rekonstruktion erforderlich ist, könnte Gaussian Splatting eine praktikable Alternative zu herkömmlichen Verfahren sein.

Eine weitere Forschungsfrage bezog sich auf die technischen Herausforderungen bei der Implementierung eines solchen Systems. Hier haben sich insbesondere die Synchronisation der Kameras, die Stabilität der API-Kommunikation mit dem MiR-Roboter und die Bildqualität der verwendeten Kameras als kritische Punkte herausgestellt. Die Arbeit zeigt, dass eine exakte Steuerung des Kamerasystems und eine präzise Missionsplanung notwendig sind, um qualitativ hochwertige Punktwolken zu erzeugen.

Insgesamt konnte die Forschung zeigen, dass Gaussian Splatting in Kombination mit günstiger Hardware eine vielversprechende Alternative für industrielle 3D-Rekonstruktionen darstellt, jedoch noch einige Verbesserungen notwendig sind, um eine durchgängig stabile und hochwertige Punktwolkengenerierung zu gewährleisten.

9.4 Möglichkeiten zur Optimierung des Workflows

Die Cloud-Integration ermöglicht eine effiziente und ressourcenschonende Verarbeitung großer Bildmengen, indem die GPU-basierte Splat-Generierung direkt nach der Aufnahme der Fotos auf leistungsstarken Servern durchgeführt wird. Anstatt die rechenintensiven Berechnungen auf einem lokalen System durchzuführen, werden die aufgenommenen Bilder automatisch in

die Cloud übertragen, wo dedizierte GPUs die Punktwolken in Gaussian Splats umwandeln. Dies reduziert die Rechenlast auf lokalen Geräten erheblich, ermöglicht eine nahezu Echtzeitverarbeitung und erleichtert die Skalierung für größere Datensätze. Dies wäre ein weiterer wichtiger Schritt zur vollständigen Automatisierung des Systems.

Der Arbeitsablauf läuft schon gut, zeigt aber Lücken bei der Datenerfassung hinsichtlich der Überlappung der Fotos. Um die Umgebung vollständig aufzunehmen, braucht es mehr Halte an den strategisch wichtigen Punkten. Alternativ kann am Ende eine extra Runde gefahren werden - dadurch ließen sich fehlende Bereiche gezielt aufnehmen. Mehr Stopps und zusätzliche Aufnahmen verbessern die 3D-Modelle und sorgen für lückenlose Punktwolken. Eine kluge Route mit Zwischenstopps, die sich an den bereits gesammelten Daten orientiert, macht den Arbeitsablauf noch effizienter.

Trotzdem muss das Kamerasystem neu evaluiert und die Positionen angepasst werden. Dies gehört jedoch nicht direkt zum Workflow, sondern zum Versuchsaufbau. Die Ergebnisse in Abbildungen 19–22 haben gezeigt, dass das System grundsätzlich funktioniert, jedoch weiter modifiziert werden muss.

Außerdem kann das Hydraulikpalettensystem des MiR genutzt werden, um das Kamera-Setup auf verschiedene Höhen zu bringen. Aktuell muss das Stativ für unterschiedliche Höhen manuell verstellt werden, um mehr Daten und damit detailliertere Informationen für die Splat-Generierung zu erfassen. Durch die Nutzung der Hydraulik könnte dieser Prozess weiter automatisiert werden, wodurch bestehende Eigenschaften des MiR optimal genutzt würden.

9.5 Weiterentwicklung durch neuronale Netze und Deep Learning

Ein neuronales Netz könnte trainiert werden, um während der Rekonstruktion Lücken in den Punktwolken zu identifizieren und gezielt weitere Aufnahmen an diesen Stellen vorzunehmen. Ebenfalls ist eine zukünftige Verbesserung der Einsatz von maschinellem Lernen zur Optimierung der Robotersteuerung. Ein Reinforcement Learning-Algorithmus könnte den optimalen Pfad basierend auf vorherigen Scans berechnen, wodurch die Missionsplanung weiter automatisiert würde.

Eine weitere Optimierungsmöglichkeit wäre der Einsatz von IMU- oder Time-of-Flight-Sensoren, um die Position des Roboters genauer zu bestimmen und Verzerrungen in der Rekonstruktion zu minimieren. Erste Studien zeigen, dass eine Kombination aus Bildern und Tiefendaten die Genauigkeit um bis zu 25 % verbessern kann [55].

9.6 Zukunftsperspektiven für Gaussian Splatting in der Industrie

Das entwickelte System zeigt großes Potenzial für eine zukünftige Open-Source-Erweiterung und kann als Alternative zu kommerziellen 3D-Scannern ausgebaut werden. Um die Konkurrenzfähigkeit zu steigern, könnten mehrere Optimierungen vorgenommen werden.

Die Integration zusätzlicher Sensoren könnte dazu beitragen, die Genauigkeit von Gaussian Splatting zu verbessern. Moderne Smartphones enthalten eine Vielzahl von Sensoren, die für eine präzisere Kamera-Positionierung genutzt werden könnten. Ein Gyroskop kann die Rotation des Geräts erfassen und zur genauen Ausrichtung der Kamera beitragen. Ein Beschleunigungssensor hilft bei der Erkennung ungewollter Bewegungen während der Aufnahme. Ein Magnetometer unterstützt die exakte Orientierung der Kamera im Raum. Ein Time-of-Flight-Sensor ermittelt Tiefeninformationen und könnte mit Gaussian Splatting kombiniert werden, um die Punktwolken-Qualität zu verbessern.

Das System wird langfristig allen Entwicklern zur Verfügung stehen - sie können es frei nutzen und für ihre Zwecke in der Industrie anpassen.

Eine interessante Perspektive für die Weiterentwicklung wäre eine Cloud-Plattform, auf der Nutzer ihre Bilder hochladen und die Berechnung extern durchführen lassen können. Dies würde die Zugänglichkeit des Systems erheblich erhöhen, insbesondere für Unternehmen, die keine High-End-GPUs besitzen.

9.7 Skalierbarkeit und Erweiterungsmöglichkeiten

Das Hinzufügen von Sensoren verbessert die Präzision beim Gaussian Splatting. Aktuelle Smartphones tragen viele Sensoren in sich, die dabei helfen, die Position der Kamera genauer zu bestimmen – durch GPS-Empfänger, Bewegungssensoren oder Beschleunigungsmesser. Dazu zählen auch Magnetometer und Gyroskope. Das Gyroskop erfasst die Rotation des Geräts und kann zur präzisen Ausrichtung der Kamera beitragen. Der Beschleunigungssensor hilft

dabei, ungewollte Bewegungen während der Aufnahme zu erkennen und auszugleichen. Ergänzend dazu unterstützt das Magnetometer die exakte Orientierung der Kamera im Raum. Ein ToF-Sensor ermöglicht die Erfassung von Tiefeninformationen und könnte in Kombination mit Gaussian Splatting dazu beitragen, die Qualität der generierten Punktwolken weiter zu verbessern.

Die Integration von ToF-Sensoren mit RGB-Kameras hat das Potenzial, die Genauigkeit der 3D-Rekonstruktion erheblich zu verbessern. Die Nutzung von ToF-Sensoren in Kombination mit RGB-Bildern führt zu hochauflösenden und präzisen Tiefenkarten, die die Genauigkeit von 3D-Raumkonstruktionen verbessern können [53]. Die Nutzung der in Smartphones integrierten Sensorik, insbesondere der LiDAR- und ToF, bietet daher eine vielversprechende Erweiterung für zukünftige Handheld-Systeme zur 3D-Rekonstruktion.

Ein Windows Mini-PC statt eines Raspberry Pi bringt mehr Stabilität ins System. Die USB-Kamera lässt sich besser ansprechen; störende Firmware-Fehler fallen eventuell weg - dabei bleibt das Gerät klein und handlich. Der schnellere Prozessor im Mini-PC erstellt die Aufnahmen zügiger. Ein Test beider Systeme zeigt, ob der Wechsel Vorteile bringt. Dabei interessiert besonders, ob das System zuverlässiger läuft und trotzdem tragbar sowie günstig bleibt.

9.7.1 Integration in andere 3D-Programme und Pipelines

Splatman, ein Tool von Colin Behrens, erweitert die Funktionalität von Gaussian Splatting, indem es eine direkte Integration in Blender ermöglicht. Dies erlaubt eine effizientere Bearbeitung und Visualisierung der Splats in einem nicht-destruktiven Workflow. Besonders für Anwendungen im Asset-Management für Games, Visual Effects und Virtual Production könnte dies eine sinnvolle Ergänzung sein. Die Nutzung innerhalb von Blender eröffnet die Möglichkeit, Splats direkt mit bestehenden 3D-Modellen zu kombinieren und an spezifische Beleuchtungssituationen anzupassen. Dies könnte vor allem für die Echtzeitvisualisierung von Punktwolken relevant sein, da Blender leistungsstarke Rendering-Engines wie Eevee oder Cycles bietet [2].

Ein sinnvoller Workflow für die Integration in Blender könnte folgendermaßen aussehen: Zunächst erfolgt die Erfassung der Punktwolke und Erstellung der Gaussian Splats in Postshot. Anschließend wird die Punktwolke in einem für Blender kompatiblen Format, beispielsweise PLY oder als EXR-Texturen für Splat-Rendering, exportiert. Danach kann sie in Blender über Splatman importiert werden, wo Anpassungen an Transparenz, Schärfe und Farbkorrektur vorgenommen werden können. Abschließend wird die Szene mit Blender-eigenen Renderern wie Cycles oder Eevee visualisiert.

Eine Integration mit Blender nützt nur bestimmten Anwendungsfeldern. Industrielle Projekte brauchen vermutlich keine Blender-Anbindung - sie kümmern sich um digitale Kopien oder um Maschinen, die gewartet werden müssen. In der Film- sowie Videoproduktion oder beim

Gestalten von Gebäuden bietet Blender dagegen neue Möglichkeiten; es ersetzt teure Programme und rendert in Echtzeit. Zukünftige Forschung wird sich vermutlich damit beschäftigen, wie Splats in Blender noch realistischer aussehen können.

Ein wichtiger nächster Schritt für das Projekt ist die Veröffentlichung der generierten Gaussian Splat-Modelle z.B. via PlayCanvas auf der Website der Fraunhofer-Gesellschaft bzw. der SmartFactoryOWL. Durch diese Integration können die erstellten 3D-Modelle direkt im Browser betrachtet und analysiert werden, ohne dass spezielle Software oder leistungsstarke Hardware erforderlich ist. PlayCanvas bietet eine webbasierte Echtzeit-Rendering-Engine, die eine interaktive Navigation durch die Punktwolken ermöglicht [54].

9.7.2 Handheld- und mobile Anwendungen

Eine Erweiterung des Systems zu einer Handheld-Variante könnte die Benutzerfreundlichkeit erheblich steigern. Während aktuelle LiDAR-basierte Handscanner hohe Kosten verursachen, könnte eine Smartphone-basierte Lösung mit Gaussian Splatting eine kostengünstige Alternative bieten [14], [15].

Ein mobiles Gerät liest Daten aus Sensoren und wertet gleichzeitig Bilder aus - so entsteht mehr Spielraum für schnelle Anpassungen. Die Informationen fließen sofort in eine automatische Bildanalyse; alle Ergebnisse synchronisieren sich direkt mit der Cloud.

9.8 Fazit und abschließende Bewertung

Basierend auf den durchgeführten Tests und der Evaluierung lässt sich festhalten, dass das entwickelte System in mehreren Bereichen eine kosteneffiziente Alternative zu bestehenden Lösungen darstellt. Besonders in Anwendungen mit begrenztem Budget kann Gaussian Splatting eine praktikable Lösung für 3D-Rekonstruktionen in Echtzeit sein.

Allerdings zeigt die Arbeit auch klare Limitationen des Systems auf. Während die Speicher- und Rechenzeiterparnis gegenüber klassischen Verfahren erheblich ist, bleibt die Gesamtqualität der Rekonstruktion stark von der Eingabebildqualität abhängig. Dies stellt insbesondere in dynamischen Umgebungen eine Herausforderung dar, in denen eine konstante Bildqualität nicht immer gewährleistet werden kann.

Dennoch zeigen die Ergebnisse, dass durch gezielte Optimierungen in der Missionsplanung, Bildverarbeitung und Sensorik das System weiterentwickelt werden kann, um eine breitere Anwendbarkeit in Industrie, Einzelhandel, Architektur und kreativen Bereichen zu ermöglichen.

Über diesen QR-Code ist eine Demo abrufbar, die die Möglichkeiten des Systems zeigen und wie die automatisierte Erstellung von Punktwolken und Gaussian Splats in einer Smartfactory aussehen kann.



V Literaturverzeichnis

- [1] The New York Times, A. J. Chavar, O. Durand und M. Boonyapanachoti, „A field guide to Gaussian splatting“, 16. Dezember 2024. <https://rd.nytimes.com/projects/gaussian-splatting-guide/>
- [2] C. Behrens und S. Strobel, „Splatman - Blender to Nerf & Gaussian Splatting“, *Blender Market*, season-04 2024. <https://blendermarket.com/products/splatman>
- [3] X. Chang, R. Zhang, J. Mao und Y. Fu, „Digital Twins in Transportation Infrastructure: An Investigation of the Key Enabling Technologies, Applications, and Challenges“, *IEEE Transactions On Intelligent Transportation Systems*, Bd. 25, Nr. 7, S. 6449–6471, Mai 2024, doi: 10.1109/tits.2024.3401716.
- [4] „NVIDIA GeForce RTX 4070 Family Graphics cards“, *NVIDIA*. <https://www.nvidia.com/en-us/geforce/graphics-cards/40-series/rtx-4070-family/>
- [5] Fuller, Z. Fan, C. Day und C. Barlow, „Digital Twin: Enabling Technologies, Challenges and Open Research“, *IEEE Access*, Bd. 8, S. 108952–108971, Jan. 2020, doi: 10.1109/access.2020.2998358.
- [6] Y. Lu, C. Liu, K. I.-K. Wang, H. Huang und X. Xu, „Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues“, *Robotics And Computer-Integrated Manufacturing*, Bd. 61, S. 101837, Aug. 2019, doi: 10.1016/j.rcim.2019.101837.
- [7] F. Tao, H. Zhang, A. Liu und A. Y. C. Nee, „Digital twin in industry: State-of-the-Art“, *IEEE Transactions On Industrial Informatics*, Bd. 15, Nr. 4, S. 2405–2415, Apr. 2019, doi: 10.1109/tii.2018.2873186.
- [8] Q.-Y. Zhou, J. Park und V. Koltun, „Open3D: A Modern Library for 3D Data Processing“, *arXiv (Cornell University)*, Jan. 2018, doi: 10.48550/arxiv.1801.09847.
- [9] B. Kerbl, G. Kopanas, T. Leimkühler und G. Drettakis, „3D Gaussian Splatting for Real-Time Radiance Field Rendering“, *arXiv (Cornell University)*, Jan. 2023, doi: 10.48550/arxiv.2308.04079.
- [10] Eigene Darstellung.
- [11] Hexagon AB (Publ), „Statistiken zu digitalen Zwillingen 2025“, *Hexagon*. <https://hexagon.com/de/resources/insights/digital-twin/statistics?utm>
- [12] Fortune Business Insights, „Digital Twin Market Size, share & Industry Analysis“, 2025. [Online]. Verfügbar unter: <https://www.fortunebusinessinsights.com/digital-twin-market-106246>
- [13] „Digital twins: Adding intelligence to the real world“, Sep. 2024. [Online]. Verfügbar unter: <https://www.capgemini.com/be-en/insights/research-library/digital-twins/>
- [14] „Matterport Kamera Produktkatalog“, *Matterport, Inc.* <https://buy.matterport.com/shop/camera/>
- [15] „Überblick zu den 3D Scannern von Faro“, *Scankraft GmbH*. https://www.scan-kraft.com/faro-3d-scanner/?srsltid=AfmBOoqxNCFeBR6iz3qJUojCos-RQk6Ksd_gFPOX_yjnelpaIzVh8cLhb
- [16] Future Market Insights Global and Consulting Pvt. Ltd., „Global 3D Laser Scanner Market Set to Reach USD 4,214 Million by 2035, Driven by Technological Advancements and

- Expanding Applications | Future Market Insights, Inc.“, *GlobeNewswire News Room*, 4. Februar 2025. [Online]. Verfügbar unter: <https://www.globenewswire.com/news-release/2025/02/04/3020102/0/en/Global-3D-Laser-Scanner-Market-Set-to-Reach-USD-4-214-Million-by-2035-Driven-by-Technological-Advancements-and-Expanding-Applications-Future-Market-Insights-Inc.html?utm>
- [17] „NVIDIA GeForce RTX 4070 Ti SUPER Benchmark Test“, *3DMark*. <https://www.3dmark.com/spy/47392606>
- [18] „Postshot System Requirements“, *Jawset Visual Computing*. <https://www.jawset.com/docs/d/Postshot%2BUser%2BGuide/Release%2BNotes/v0.1?utm>
- [19] Zanolari, „Open Source Software für KMU – Vorteile, Chancen, Risiken“, *Topsoft*, 15. Februar 2024. <https://topsoft.ch/themen/open-source-software-fuer-kmu-vorteile-chancen-risiken/?utm>
- [20] F. Merschel, „Erstellung und Darstellung von Gaussian Splats - Optimierung des Arbeitsablaufs, der Fotobearbeitung und Floater Minimierung“, Technische Hochschule Ostwestfalen-Lippe, 2025.
- [21] C. Changjian Jiang, R. Gao, K. Shao, Y. Wang, R. Xiong und Y. Zhang, „LI-GS: Gaussian Splatting with LiDAR Incorporated for Accurate Large-Scale Reconstruction“, Sep. 2024. [Online]. Verfügbar unter: <https://arxiv.org/html/2409.12899v1?>
- [22] D. Robert, „Efficient learning on large-scale 3D point clouds“, Doktorarbeit, Université Gustave Eiffel, 2024. [Online]. Verfügbar unter: <https://theses.hal.science/tel-04615714v1>
- [23] F. Schmidt, M. Enzweiler und A. Valada, „NeRF and Gaussian Splatting SLAM in the Wild“, *arXiv (Cornell University)*, Dez. 2024, doi: 10.48550/arxiv.2412.03263.
- [24] M. Jarofka, S. Schweig, N. Maas und D. Schramm, „Toolchain Development for Automated Scene Reconstruction using Artificial Neural Network Object Detection and Photogrammetry for the Application in Driving Simulators“, Jan. 2020. doi: 10.5220/0009590500250034., S. 25–34
- [25] N. Snavely, S. M. Seitz und R. Szeliski, „Modeling the World from Internet Photo Collections“, *International Journal Of Computer Vision*, Bd. 80, Nr. 2, S. 189–210, Dez. 2007, doi: 10.1007/s11263-007-0107-3.
- [26] National Oceanic and Atmospheric Administration (NOAA), „What is lidar?“, *National Ocean Service (NOAA)*, 16. Juni 2024. <https://oceanservice.noaa.gov/facts/lidar.html?>
- [27] M.-C. Papen, K. M. Magalhaes, S. Tenbrock und C. Märkel, „Digitalisierungsanwendungen und Identifikation von Digitalisierungstrends im Mittelstand“, 2022. <https://www.econstor.eu/handle/10419/267680?>
- [28] Y.-J. Yuan, Y.-T. Sun, Y.-K. Lai, Y. Ma, R. Jia und L. Gao, „NeRF-Editing: Geometry Editing of Neural Radiance Fields“, *2022 IEEE/CVF Conference On Computer Vision And Pattern Recognition (CVPR)*, S. 18332–18343, Juni 2022, doi: 10.1109/cvpr52688.2022.01781.
- [29] K. Gao, Y. Gao, H. He, D. Lu, L. Xu und J. Li, „NERF: Neural Radiance Field in 3D Vision, A Comprehensive review“, *arXiv.org*, 1. Oktober 2022. <https://arxiv.org/abs/2210.00379v5>

- [30] Goesele *u. a.*, „Multi-View Stereo for Community Photo Collections“. [Online]. Verfügbar unter: <https://grail.cs.washington.edu/projects/mvscpc/download/Goesele-2007-MVS.pdf>, S. 1-8
- [31] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi und R. Ng, „NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis“, *arXiv.org*, 19. März 2020. <https://arxiv.org/abs/2003.08934>
- [32] C. Sun, M. Sun und H.-T. Chen, „Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction“, *arXiv (Cornell University)*, Jan. 2021, doi: 10.48550/arxiv.2111.11215.
- [33] G. Chen und W. Wang, „A Survey on 3D Gaussian Splatting“, *arXiv (Cornell University)*, Jan. 2024, doi: 10.48550/arxiv.2401.03890.
- [34] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin und A. Geiger, „Occupancy Networks: Learning 3D reconstruction in function space“, *2022 IEEE/CVF Conference On Computer Vision And Pattern Recognition (CVPR)*, Juni 2019, doi: 10.1109/cvpr.2019.00459.
- [35] J. J. Park, P. Florence, J. Straub, R. Newcombe und S. Lovegrove, „DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation“, *2022 IEEE/CVF Conference On Computer Vision And Pattern Recognition (CVPR)*, S. 165–174, Juni 2019, doi: 10.1109/cvpr.2019.00025.
- [36] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht und A. Kanazawa, „K-Planes: Explicit Radiance Fields in Space, Time, and Appearance“, *2022 IEEE/CVF Conference On Computer Vision And Pattern Recognition (CVPR)*, S. 12479–12488, Juni 2023, doi: 10.1109/cvpr52729.2023.01201.
- [37] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan und P. Hedman, „MIP-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields“, *arXiv (Cornell University)*, Jan. 2021, doi: 10.48550/arxiv.2111.12077.
- [38] X. Fang, Y. Zhang, H. Tan, C. Liu und X. Yang, „Performance Evaluation and Optimization of 3D Gaussian Splatting in Indoor Scene Generation and Rendering“, *ISPRS International Journal Of Geo-Information*, Bd. 14, Nr. 1, S. 21, Jan. 2025, doi: 10.3390/ijgi14010021.
- [39] Z. Peng, Y. Yang, T. Shao, C. Jiang und K. Zhou, „X-SLAM: Scalable Dense SLAM for Task-aware Optimization using CSFD“, *ACM Transactions On Graphics*, Bd. 43, Nr. 4, S. 1–15, Juli 2024, doi: 10.1145/3658233.
- [40] R. Zha, T. J. Lin, Y. Cai, J. Cao, Y. Zhang und H. Li, „R²-Gaussian: Rectifying Radiative Gaussian Splatting for Tomographic Reconstruction“, *arXiv (Cornell University)*, Mai 2024, doi: 10.48550/arxiv.2405.20693.
- [41] „GeForce RTX 4090 vs RTX 4070: Tests and Benchmarks“, *NanoReview.net*. <https://nanoreview.net/en/gpu-compare/geforce-rtx-4090-vs-geforce-rtx-4070>
- [42] „Topic: Optimizing Processing Time?“, *Agisoft*. https://www.agisoft.com/forum/index.php?topic=15321.0&utm_source=chatgpt.com
- [43] AnkerWork, „PowerConf C200 Webcam“, *AnkerWork*. <https://de.ankerwork.com/products/a3369>
- [44] „Logitech BRIO Webcam mit 4K-Ultra-HD-Video & HDR“. <https://www.logitech.com/de-de/products/webcams/brio-4k-hdr-webcam.html>

- [45] „Raspberry Pi 4 B“, *Reichelt Elektronik GmbH*. https://www.reichelt.de/de/de/shop/produkt/raspberry_pi_4_b_4x_1_5_ghz_4_gb_ram_wlan_bt-259920
- [46] „MiR600 EU Pallet Lift“, *Mobile Industrial Robots A/S*. <https://mobile-industrial-robots.com/de/produkte/anwendungen/mir-eu-pallet-lift-600>
- [47] „Raspberry Pi - Kamera“, *Reichelt Elektronik GmbH*. https://www.reichelt.de/de/de/shop/produkt/raspberry_pi_-_kamera_12mp_75_c_-_cs-fassung-276919?PROVID=2788&gQT=1
- [48] „Raspberry Pi - Objektiv“, *Reichelt Elektronik GmbH*. https://www.reichelt.de/de/de/shop/produkt/raspberry_pi_-_objektiv_fuer_cs-fassung_6mm_weitwinkel-276922
- [49] „3D-Scanner auf der Baustelle“, *Drees & Sommer Schweiz AG*. <https://www.dreso.com/ch/top-themen/bim-solutions/3d-scanner-baustelle>
- [50] H. Schwirzer, „Baufaufnahme mit Laserscanner“, *Baufuchs*. <https://www.baufuchs.com/planung/3d-laserscanner/>
- [51] KreativInstitut.Ostwestfalen-Lippe, „Gaussian Splat: Hermannsdenkmal“, *KreativInstitut.OWL*. <https://kreativ.institute/de/projekte-und-beitraege/gaussian-splat/hermannsdenkmal>
- [52] Y. Chen, M. Tsukada und H. Esaki, „Reinforcement Learning Based Optimal Camera Placement for Depth Observation of Indoor Scenes“, *arXiv (Cornell University)*, Jan. 2021, doi: 10.48550/arxiv.2110.11106.
- [53] Y. Li u. a., „DELTA: Depth Estimation from a Light-Weight ToF Sensor and RGB Image“, in *Lecture notes in computer science*, 2022, S. 619–636. doi: 10.1007/978-3-031-19769-7_36.
- [54] W. Eastcott, „Create 3D Gaussian Splat Apps with the PlayCanvas Editor“, *PlayCanvas Blog*, 5. Juni 2024. <https://blog.playcanvas.com/create-3d-gaussian-splat-apps-with-the-playcanvas-editor/>
- [55] S. Manthe, „3D-Rekonstruktion aus monokularen Bilderserien“, 2014. [Online]. Verfügbar unter: https://kola.opus.hbz-nrw.de/frontdoor/deliver/index/docId/891/file/bachelorarbeit_stephan_manthe.pdf
- [56] Z. Liao u. a., „Fisheye-GS: Lightweight and Extensible Gaussian Splatting Module for Fisheye Cameras“, *arXiv (Cornell University)*, Sep. 2024, doi: 10.48550/arxiv.2409.04751.
- [57] G. Zhekov, „3D Gaussian Splatting: A new frontier in rendering“. <https://www.chaos.com/blog/3d-gaussian-splatting-new-frontier-in-rendering>
- [58] S. Szymanowicz, C. Rupprecht und A. Vedaldi, „Splatter Image: Ultra-Fast Single-View 3D reconstruction“, *arXiv (Cornell University)*, Jan. 2023, doi: 10.48550/arxiv.2312.13150.
- [59] F. Warburg, E. Weber, M. Tancik, A. Holynski und A. Kanazawa, „Nerfbusters: Removing Ghostly Artifacts from Casually Captured NeRFs“, *arXiv (Cornell University)*, Jan. 2023, doi: 10.48550/arxiv.2304.10532.
- [60] V. Yugay, Y. Li, T. Gevers und M. R. Oswald, „Gaussian-SLAM: Photo-realistic Dense SLAM with Gaussian Splatting“, *arXiv (Cornell University)*, Jan. 2023, doi: 10.48550/arxiv.2312.10070.

VI Anhang

```
1  #!/bin/bash
2
3  # Base URL and Authorization Header
4  BASE_URL="http://172.16.168.49/api/v2.0.0/mission_queue"
5  AUTH_HEADER="Authorization: Basic
6  UmVzdEFQSTowMTIwZGE2MDIwYWVmZjFmODk0ODVlYzA0MTZjZmFmZGVmYjJlNzNlNW4MmUzYTJhZjhjMGY1OTIz:
7  LANG_HEADER="Accept-Language: en_US"
8  MISSION_ID="mirconst-guid-0000-0003-actionlist00"
9  FTP_SCRIPT="/home/sam/capture_to_ftp.sh"
10 STATUS_URL="http://172.16.168.49/api/v2.0.0/status"
11
12 # 1) Enqueue mission to set footprint (SetFootprint_GaussianSplatting)
13 echo "Enqueuing SetFootprint_GaussianSplatting mission..."
14 curl -X POST "$BASE_URL" \
15     -H "$AUTH_HEADER" \
16     -H "accept: application/json" \
17     -H "$LANG_HEADER" \
18     -H "Content-Type: application/json" \
19     -d '{
20         "mission_id": "301c06e1-d8d8-11ef-bd8d-000e8e987f45",
21         "parameters": [],
22         "priority": 1
23     }'
24 echo "Set footprint mission queued."
25
26 # Array of Waypoints
27 WAYPOINTS=(
28     "a195076b-d72b-11ef-bd8d-000e8e987f45:-8.546:18.499:0.0"
29     "bcc3f6b2-d72b-11ef-bd8d-000e8e987f45:20.618:18.247:-90.0"
30     "c983608f-d72b-11ef-bd8d-000e8e987f45:20.753:3.42:90.0"
31     "d6a90c2a-d72b-11ef-bd8d-000e8e987f45:21.157:18.179:179.0"
32     "e5b725e5-d72b-11ef-bd8d-000e8e987f45:11.25:18.247:-90.0"
33     "f82b52ce-d72b-11ef-bd8d-000e8e987f45:11.344:5.594:-180.0"
34     "04c89027-d72c-11ef-bd8d-000e8e987f45:-3.891:4.655:1.0"
35     "0f8e6bc4-d72c-11ef-bd8d-000e8e987f45:6.179:5.036:90.0"
36     "1c7aaaab-d72c-11ef-bd8d-000e8e987f45:6.061:18.247:179.0"
37 )
38
39 # Helper function to calculate distance
40 calculate_distance() {
41     local x1=$1
42     local y1=$2
43     local x2=$3
44     local y2=$4
45     echo "scale=2; sqrt(($x2 - $x1)^2 + ($y2 - $y1)^2)" | bc
46 }
47
48 # Main Loop: Move Between Waypoints
49 for ((i=0; i<${#WAYPOINTS[@]}-1; i++); do
50     # Extract current and next waypoint details
51     IFS=":" read -r GUID1 X1 Y1 ORIENTATION1 <<< "${WAYPOINTS[i]}"
52     IFS=":" read -r GUID2 X2 Y2 ORIENTATION2 <<< "${WAYPOINTS[i+1]}"
53
54     # Calculate total distance
55     TOTAL_DISTANCE=$(calculate_distance "$X1" "$Y1" "$X2" "$Y2")
56     STEPS=$(echo "$TOTAL_DISTANCE / 1" | bc)
```

```

56
57     echo "Moving from Waypoint_$(i+1) to Waypoint_$(i+2) in $STEPS steps."
58
59     for ((step=1; step<=STEPS; step++)); do
60         # Calculate incremental position
61         FRACTION=$(echo "scale=2; $step / $STEPS" | bc)
62         X_CURRENT=$(echo "$X1 + $FRACTION * ($X2 - $X1)" | bc)
63         Y_CURRENT=$(echo "$Y1 + $FRACTION * ($Y2 - $Y1)" | bc)
64
65         # Create unique fleet schedule GUID
66         TIMESTAMP=$(date +%s%N)
67
68         # Create JSON payload
69         PAYLOAD=$(cat <<EOF
70 {
71     "priority": 1,
72     "description": "Move to incremental position ($X_CURRENT, $Y_CURRENT)",
73     "parameters": [
74         {"id": "X", "value": $X_CURRENT, "label": "X"},
75         {"id": "Y", "value": $Y_CURRENT, "label": "Y"},
76         {"id": "Orientation", "value": $ORIENTATION1, "label": "Orientation"}
77     ],
78     "fleet_schedule_guid": "$TIMESTAMP",
79     "mission_id": "$MISSION_ID",
80     "message": ""
81 }
82 EOF
83 )
84
85     # Send request to move the robot
86     RESPONSE=$(curl -s -o /dev/null -w "%{http_code}" -X POST "$BASE_URL" \
87         -H "accept: application/json" \
88         -H "$AUTH_HEADER" \
89         -H "$LANG_HEADER" \
90         -H "Content-Type: application/json" \
91         -d "$PAYLOAD")
92
93     if [ "$RESPONSE" -eq 201 ]; then
94         echo "Robot moved to position ($X_CURRENT, $Y_CURRENT)."
95     else
96         echo "Failed to move to position ($X_CURRENT, $Y_CURRENT). Response code: $RESPONSE"
97         continue
98     fi
99
100    # Wait for the robot to stabilize
101    sleep 3
102
103    # Check if the robot is stationary
104    while true; do
105        ROBOT_STATE=$(curl -s -X GET "$STATUS_URL" \
106            -H "accept: application/json" \
107            -H "$AUTH_HEADER" | jq -r '.velocity.linear')
108
109        if (( $(echo "$ROBOT_STATE == 0" | bc -l) )); then
110            echo "Robot is stationary. Triggering cameras."
111            if bash "$FTP_SCRIPT"; then
112                echo "Cameras triggered successfully."
113            else
114                echo "Failed to trigger cameras."
115            fi
116            break
117        fi
118        sleep 1
119    done

```

```
119 done
120 done
121 done
122
123 echo "All waypoint transitions completed."
124
125 sleep 10
126 ./delete_mission.sh
127
128 # 2) Enqueue mission to reset footprint
129 echo "Enqueuing ResetFootprint_GaussianSplatting mission..."
130 curl -X POST "$BASE_URL" \
131   -H "$AUTH_HEADER" \
132   -H "accept: application/json" \
133   -H "$LANG_HEADER" \
134   -H "Content-Type: application/json" \
135   -d '{
136     "mission_id": "a9f2ae8b-e3e7-e0fb-5dff-276c0acc5950",
137     "parameters": [],
138     "priority": 1
139   }'
140 echo "Reset footprint mission queued."
141
```

Abbildung 23: "generate_mission.sh"- Skript zur Erstellung der Missionswege des MiR


```
1  #!/bin/bash
2
3  # Directory for FTP uploads (local on the Raspberry Pi)
4  FTP_DIR="/home/sam/FTP/files"
5
6  # Ensure the directory exists
7  mkdir -p $FTP_DIR
8
9  # Camera list
10 cameras="/dev/video0" "/dev/video4" "/dev/video8" "/dev/video25"
11
12 # Desired resolution
13 resolution="3840x2160"
14
15 # Loop through each camera and capture an image
16 for cam in "${cameras[@]}; do
17     # Create a unique filename with timestamp and camera ID
18     filename="$(basename $cam)_$(date +%Y%m%d_%H%M%S).jpg"
19     filepath="$FTP_DIR/$filename"
20
21     # Capture the image
22     echo "Capturing image from $cam..."
23     fswebcam --no-banner -d "$cam" -r "$resolution" "$filepath"
24
25     echo "Image saved for $cam: $filepath"
26 done
27
28 echo "All images captured and saved in $FTP_DIR."
29
```

Abbildung 24: "capture_to_fip.sh"- Skript zum Triggern der Kameras und Speicherung dieser auf dem FTP Server mit einem Timestamp als Namen

```
1  pkill -f generate_mission.sh
2
3  curl -X DELETE "http://172.16.168.49/api/v2.0.0/mission_queue" -H "accept: application/json" -t
  "Authorization: Basic
  UmVzdeFQSTowMTIwZGE2MDIwYWNmZjFmODk0ODVLYzA0MTZjZmFmZGVmYjJlNzNlNWw4MmUzYTJhZjhjMGY1OTIzZjFLZTE
  -H "Accept-Language: en_US"
4
5
```

Abbildung 25: "delete_mission..sh"- Skript zum Löschen aller Wegpunkte und dem Stop des MiR

```
1 #!/bin/bash
2
3 # List of camera devices
4 CAMERAS="/dev/video0" "/dev/video4" "/dev/video8" "/dev/video25"
5
6 for cam in "${CAMERAS[@]"; do
7     echo "Setting controls for $cam.."
8
9 # (Optional) Set the video format/resolution
10 # Adjust pixelformat if needed (e.g., MJPG, YUYV, etc.).
11 sudo v4l2-ctl -d "$cam" --set-fmt-video=width=3840,height=2160,pixelformat=MJPG
12
13 # User Controls
14 sudo v4l2-ctl -d "$cam" --set-ctrl=brightness=128
15 sudo v4l2-ctl -d "$cam" --set-ctrl=contrast=128
16 sudo v4l2-ctl -d "$cam" --set-ctrl=saturation=128
17 sudo v4l2-ctl -d "$cam" --set-ctrl=white_balance_automatic=1
18 sudo v4l2-ctl -d "$cam" --set-ctrl=gain=0
19 sudo v4l2-ctl -d "$cam" --set-ctrl=power_line_frequency=1
20 sudo v4l2-ctl -d "$cam" --set-ctrl=white_balance_temperature=4050
21 sudo v4l2-ctl -d "$cam" --set-ctrl=sharpness=185
22 sudo v4l2-ctl -d "$cam" --set-ctrl=backlight_compensation=1
23
24 # Camera Controls
25 sudo v4l2-ctl -d "$cam" --set-ctrl=auto_exposure=1
26 sudo v4l2-ctl -d "$cam" --set-ctrl=exposure_time_absolute=1000
27 sudo v4l2-ctl -d "$cam" --set-ctrl=exposure_dynamic_framerate=0
28 sudo v4l2-ctl -d "$cam" --set-ctrl=pan_absolute=0
29 sudo v4l2-ctl -d "$cam" --set-ctrl=tilt_absolute=-14400
30 sudo v4l2-ctl -d "$cam" --set-ctrl=focus_absolute=8
31 sudo v4l2-ctl -d "$cam" --set-ctrl=focus_automatic_continuous=0
32 sudo v4l2-ctl -d "$cam" --set-ctrl=zoom_absolute=100
33 sudo v4l2-ctl -d "$cam" --set-ctrl=led1_mode=3
34 sudo v4l2-ctl -d "$cam" --set-ctrl=led1_frequency=20
35
36 echo "Done setting $cam."
37 done
38
39 echo "All cameras updated."
40
```

Abbildung 26: `set_camera.sh`- Skript zum Bestätigen der Kameraeinstellungen, gerade hilfreich um alle Kameras zu synchronisieren

VII Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Alle sinngemäß und wörtlich übernommenen Textstellen aus fremden Quellen wurden kenntlich gemacht.

Lage, den 10.02.2025

A handwritten signature in black ink, consisting of a stylized 'S' and 'W' with a horizontal line underneath.

Sam Wiemann