

SCHUTZWERK



Masterarbeit

Studiengang Informatik

Automatisierung der hardwarenahen Sicherheitsanalyse von eingebetteten Systemen

Fabian Weber

Aufgabensteller/Prüfer Prof. Dr. Elmar Böhler
Zweitprüfer Prof. Dr. rer. nat. Stefan Frenz
Arbeit vorgelegt am 16. September 2019

durchgeführt bei SCHUTZWERK GmbH
Betreuer Pfarrer-Weiß-Weg 12
Arbeit vorgelegt am 16. September 2019
Dr. Bastian Königs

Fassung vom: 16. September 2019



Dieses Werk ist unter einer Creative Commons Lizenz vom Typ
Namensnennung 4.0 International zugänglich.

Um eine Kopie dieser Lizenz einzusehen, konsultieren Sie
<https://creativecommons.org/licenses/by/4.0/deed.de>
oder wenden Sie sich brieflich an Creative Commons, PO Box 1866,
Mountain View, CA 94042, USA.

Abstract

Durch neuere Entwicklungstrends, wie das Internet der Dinge (*Internet of Things (IoT)*) oder die zunehmende Vernetzung von Industrie- und Fertigungsanlagen (Industrie 4.0), finden immer mehr elektronische Geräte Einzug in das alltägliche Leben. Aufgrund ihrer starken Verbreitung und Vernetzung bieten diese eine große Angriffsfläche hinsichtlich der IT-Sicherheit. Forschungen bezüglich der automatisierten Sicherheitsanalyse solcher eingebetteten Systeme fokussieren sich vor allem auf die enthaltenen Softwarekomponenten, sodass die elektronische Analyse auf Platinenebene bisher größtenteils manuell durchgeführt werden muss. Motiviert von dieser Tatsache wurde eine modulare Hardwareplattform für die automatisierte IT-Sicherheitsanalyse eingebetteter Systeme entwickelt und umgesetzt. Der aufgebaute Prototyp erlaubt dazu eine automatisierte, elektrische Kontaktierung mit bis zu zwei Prüfspitzen. Durch ein integriertes, fahrbares Kamerasystem können hochaufgelöste Aufnahmen des Untersuchungsobjekts aufgenommen werden. Weiterhin wurde eine Kalibrierungsroutine entwickelt, die eine Kopplung des Kamerasystems mit den Prüfspitzen erlaubt. Auf einer Testplatine konnte mit dem zuvor kalibrierten, gekoppelten System eine Positionsgenauigkeit von $\pm 0,1$ mm in der Ebene erzielt werden. Neben elektrischen Tests, sind durch das integrierte Kamerasystem auch visuelle Analysen eines *Printed Circuit Boards (PCBs)* möglich. Hierfür wurden automatisierte Verfahren basierend auf Neuronalen Netzen zur Erkennung von sicherheitsrelevanten Komponenten und einzelnen Bauteilpins evaluiert. Mit den untersuchten Modellen konnte eine *Average Precision (AP)* von bis zu 87,4 % bei der Komponentenerkennung auf dem eingesetzten Evaluationsdatensatz erzielt werden. Bauteilpins wurden mit einer größten gemessenen *mean Precision (mP)* von 86,3 % bei einem *mean Recall (mR)* von 77,9 % erkannt. Für die teilautomatisierte, interaktive Ansteuerung der entwickelten Hardwareplattform wurde eine Software mit grafischer Benutzeroberfläche implementiert, die einem Analysten die manuelle Selektion von Testpunkten ermöglicht. Es wurde somit eine Hard- und Softwareplattform geschaffen, welche als Basis für weitere Untersuchungen automatisierter Testverfahren genutzt werden kann.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Problemstellung	1
1.2. Zielsetzung	3
1.3. Beiträge und Aufbau der Arbeit	4
2. Visuelle PCB-Analyse	6
2.1. Verwandte Arbeiten	7
2.2. Grundlagen	9
2.2.1. Neuronale Netze zur Objekterkennung	10
2.2.2. Die Metrik der <i>Pascal VOC Challenge</i>	13
2.2.3. Bildsegmentierung mittels globaler Schwellwertbildung	14
2.2.4. Morphologische Operationen	15
2.2.5. Floodfill	16
2.2.6. Bilaterale Filterung	16
2.2.7. Distance Transform	17
2.2.8. Zusammenhangskomponenten	17
2.3. Chiperkennung mit Neuronalen Netzen	17
2.3.1. Verwendete Modelle	18
2.3.2. Bilddatensätze	18
2.3.3. Training	20
2.3.4. Evaluation	23
2.4. Pin-Erkennung	25
2.4.1. Bilddatensatz	27
2.4.2. Computer Vision Pipeline zur Pinerkennung	27
2.4.3. Pinerkennung mit Neuronalen Netzen	31
2.4.4. Evaluation	31
2.5. Zusammenfassung und Ausblick	35
3. Hardwareplattform für die automatisierte PCB-Kontaktierung	38
3.1. Stand der Technik	39
3.2. Anforderungen	40
3.2.1. Mechanische Anforderungen	41

3.2.2. Anforderungen an das Kamerasytem	42
3.3. Ansatz	43
3.3.1. Mögliche mechanische Plattformen zur PCB-Kontaktierung	43
3.3.2. Mögliche Varianten eines Kamerasytems	45
3.3.3. Auswahl der Teilsysteme	46
3.4. Entwickelte Hardware-Plattform	47
3.4.1. Kinematik	48
3.4.2. Linearführungssysteme	49
3.4.3. Antriebe	52
3.4.4. Softwareseitige Ansteuerung	56
3.4.5. Finaler Prototyp	58
3.4.6. Beobachtungen	62
3.5. Zusammenfassung und Ausblick	64
4. Kalibrierung der Hardwareplattform	67
4.1. Stand der Technik	67
4.2. Grundlagen	68
4.2.1. Das Kameramodell der Computer Vision	69
4.2.2. Kreis-Fitting	70
4.2.3. Koordinatentransformation	71
4.2.4. Bilineare Interpolation	74
4.3. Eigener Ansatz	75
4.4. Kalibrierung der Kontaktiereinheiten	76
4.5. Kalibrierung des bewegten Kamerasytems	79
4.6. Evaluation	85
4.6.1. Methodik	85
4.6.2. Ergebnisse	88
4.7. Image Stitching zur Generierung hochauflöster PCB-Aufnahmen	91
4.7.1. Eingesetztes Verfahren	91
4.7.2. Evaluation	92
4.8. Zusammenfassung und Ausblick	94
5. Zusammenfassung	97
Anhang	V
A. Trainingsverlauf der Neuronalen Netze zur Chiperkennung	V
B. Trainingsverlauf der Neuronalen Netze zur Pinerkennung	VII
C. Motorkennline Schrittmotor Nanotec - ST3518M1004-A	IX

D. Entwürfe der Kontaktierhardware

X

Literaturverzeichnis

XIII

Bildquellen

XVIII

1. Einleitung

Neue Entwicklungen wie das *Internet of Things* (IoT) und Industrie 4.0 haben zu einer vermehrten Entwicklung von kleinen, verteilten Rechnersystemen geführt. In vielen Fällen kommen dabei sogenannte *eingebettete Systeme* zum Einsatz. Dabei handelt es sich um Rechnersysteme, die in einen physikalischen Kontext eingebettet sind und mit diesem interagieren. Um ihre Aufgabe möglichst effizient zu erfüllen, ist die Soft- und auch Hardware solcher Systeme meist speziell auf die jeweiligen Anforderungen abgestimmt. Zur Erfüllung ihrer Funktion müssen diese Systeme in vielen Fällen mit weiteren Rechnersystemen interagieren. Sie sind daher oftmals über Kommunikationssysteme wie dem Internet oder lokalen Kommunikationsbussen mit anderen Rechnersystemen verbunden.

Durch die zunehmende Verbreitung und Vernetzung bieten eingebettete Systeme für potentielle Angreifer eine beträchtliche Angriffsfläche [1]. Daher rückt vor allem in sicherheitskritischen Bereichen, wie Industrieanlagen oder Fahr- und Flugzeugen, die Sicherheit solcher Systeme bezüglich der Angreifbarkeit und Manipulierbarkeit in den Fokus bei der Entwicklung. Analysen, um zuverlässige Aussagen bezüglich der Sicherheit zu treffen, gestalten sich aufgrund des individuellen und meist hochkomplexen Aufbaus oftmals sehr schwierig und sind nur von Experten durchführbar.

Im Automobilbereich kommen zahlreiche eingebettete Systeme in unterschiedlichen Bereichen zum Einsatz. Sie werden beispielsweise in Infotainment Systemen eingesetzt, übernehmen aber auch sicherheitsrelevante Funktionen, wie etwa die Motorsteuerung oder Auswertungen von Sensorsignalen zur Unfallerkennung. Auch bei neueren Entwicklungen hin zum autonomen Fahren spielen eingebettete Systeme eine zentrale Rolle. Rechnersysteme übernehmen dabei die vollständige Kontrolle über das Fahrzeug. Ein erfolgreicher Angriff auf solche Systeme wäre gravierend. Automobilhersteller beschäftigen sich daher intensiv mit dem Thema der IT-Sicherheit in diesem Kontext. Auch das Bundesministerium für Bildung und Forschung (BMBF) unterstützt die Forschung auf diesem Gebiet durch das Förderprojekt SecForCARS [2].

1.1. Problemstellung

Die Sicherheitsanalyse eingebetteter Systeme ist derzeit ein überwiegend manuell durchgeführter Prozess¹. Falls nicht explizit genannt, wird im Weiteren immer von einem sogenannten

¹Ergebnis eines Interviews mit Sicherheitsanalysten für den Bereich *Embedded Systems* der SCHUTZWERTK GmbH am 05.03.2019

Black Box-Test eines eingebetteten Systems ausgegangen. Bei einem solchen Test besitzt der Sicherheitsanalyst keine Vorkenntnisse über das zu testende System. Er befindet sich daher in derselben Situation wie ein realer Angreifer. Es wird weiterhin angenommen, dass sich die durchgeführte Sicherheitsanalyse auf ein einzelnes, isoliertes, eingebettetes System beschränkt und sich alle IT-sicherheitsrelevanten Komponenten auf einer einzelnen Leiterplatte befinden.

Nachfolgend wird der Ablauf einer manuellen Sicherheitsanalyse eines eingebetteten Systems beschrieben, wie er bei der *SCHUTZWERK GmbH* derzeit eingesetzt wird. Der vorgestellte Prozess ist stark vereinfacht und soll lediglich ein Grundverständnis für die notwendigen Schritte einer Sicherheitsanalyse schaffen.

Nach der physischen Isolierung des eingebetteten Systems durch Entfernung vorhandener Gehäuseteile und weiterer Peripherie wird zunächst eine visuelle Analyse der Leiterplatte durchgeführt. Diese wird auch als *Printed Circuit Board* (PCB) bezeichnet. Im Weiteren wird der Begriff PCB synonym für eingebettete Systeme verwendet. Bei der visuellen Analyse verschafft sich ein Sicherheitsanalyst zunächst einen Überblick über alle auf dem PCB vorhandenen Bauteile und wie diese untereinander verbunden sind. Ziel hierbei ist, ein grobes Verständnis des PCBs und dessen Funktionsweise zu erlangen. Meist erfolgt in diesem Schritt bereits die Identifikation einzelner Bauteile, was die Grundlage für die weiteren Analyseschritte bildet. Bis hierher entspricht das Vorgehen dem *Reverse Engineering* des zu untersuchenden PCBs. Beim *Reverse Engineering* eines PCBs wird versucht, dessen ursprüngliche Konstruktionsdaten anhand einer vorliegenden, vollständig bestückten Platine zu rekonstruieren [3, S. 19].

Mit den gesammelten Informationen werden anschließend potentielle Schwachstellen gesucht. Im Fokus liegen hierbei vor allem aktive Bausteine, wie etwa Mikrocontroller, oder Informationsspeicher in Form externer Speicherbausteine. Diese bieten einem Angreifer die größte Angriffsfläche zur Manipulation des Systems oder der Extraktion sensibler, in dem System gespeicherter Daten.

Mittels einer elektrischen Kontaktierung des PCBs beginnt der Sicherheitsanalyst dann mit der Analyse der potentiellen Schwachstellen. Die elektrische Kontaktierung erfolgt etwa über angebrachte Klemmverbindungen, häufig aber, aufgrund der oftmals sehr kleinen Bausteine, über händisch platzierte Kontaktiernadeln und angelötete Kabel. Die Anbringung dieser elektrischen Verbindungen ist sehr zeitaufwendig und fehleranfällig. Durch Unachtsamkeiten kann das zu untersuchende System schnell beschädigt oder sogar zerstört werden.

Ist die physikalische Verbindung zum Untersuchungsobjekt hergestellt, beginnt der Sicherheitsanalyst mit der eigentlichen Analyse. Dabei werden etwa Daten von Speicherbausteinen extrahiert und analysiert oder Penetrationstests durchgeführt. Bei letzteren handelt es sich um Tests, bei denen versucht wird, die Kontrolle über das System zu erlangen oder gezielt Informationen auszulesen. Dazu werden etwa exponierte Debugging- oder Kommunikationsschnittstellen zur Interaktion mit dem System genutzt.

Gefundene Schwachstellen werden dokumentiert und an den Hersteller des Systems übermittelt. Sie fließen in den Entwicklungsprozess mit ein und resultieren, bei Vorliegen gravierender

Schwachstellen, meist in einer neuen, verbesserten Revision des Produktes. Die Abfolge von Sicherheitsanalyse und Produktverbesserung wird solange wiederholt, bis keine Sicherheitslücken mehr gefunden wurden oder das verbliebene Restrisiko als entsprechend gering erachtet wird. Ein wiederholter Test mit einer neueren Produktversion wird auch als *Retest* bezeichnet. Bei einem solchen kann der Sicherheitsanalyst zwar auf Ergebnisse früherer Tests zurückgreifen, die aufwendigen Schritte der elektrischen Kontaktierung müssen aber für jedes Untersuchungsobjekt erneut durchgeführt werden.

1.2. Zielsetzung

Die vorliegende Arbeit beschäftigt sich generell mit der Automatisierung des im vorigen Kapitel beschriebenen Prozesses der manuellen Sicherheitsanalyse eingebetteter Systeme. Sie orientiert sich dabei an dem in Abbildung 1.1 dargestellten Entwurf eines äquivalenten, automatisierten Prozessablaufs. Dieser wurde anhand der Beschreibung des derzeit bei der *SCHUTZWERK GmbH* eingesetzten Vorgehens entwickelt. Dabei konzentriert sich die Arbeit auf die in Abbildung 1.1 blau eingefärbten Prozessschritte: Die visuelle PCB-Analyse und eine automatisierte, elektrische Kontaktierung.

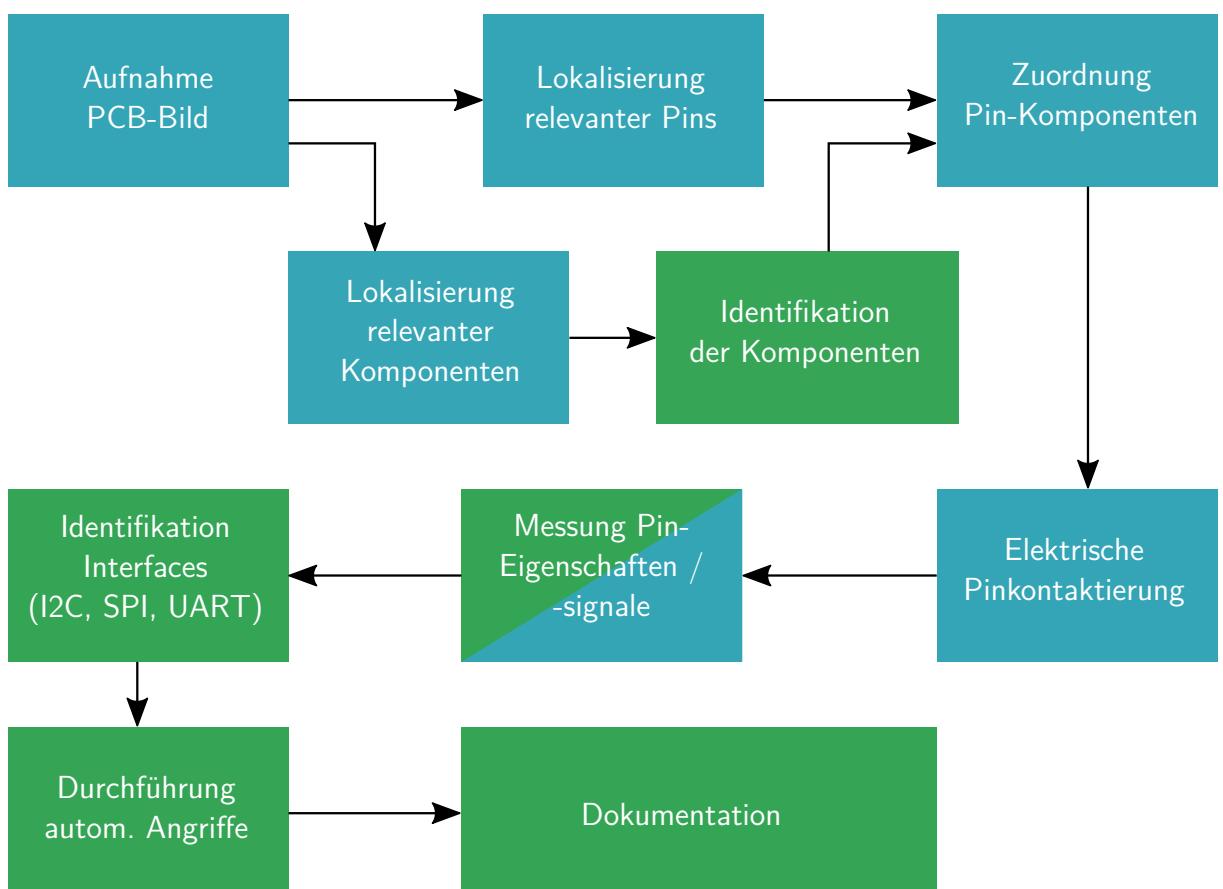


Abbildung 1.1.: Entwurf eines automatisierten Prozesses zur Sicherheitsanalyse eingebetteter Systeme; blau eingefärbte Prozessschritte sind Gegenstand dieser Arbeit

Für die Automatisierung der visuellen Analyse von PCBs sollen neue, dem aktuellen Stand der Technik entsprechende Verfahren untersucht werden. Ziel dabei ist die möglichst robuste und zuverlässige Identifizierung und Lokalisierung sicherheitsrelevanter Bauteile. Weiterhin soll eine Methodik entwickelt werden, die eine präzise Lokalisierung von Bauteilpins ermöglicht. Die so gewonnenen Informationen bilden den Ausgangspunkt für das Kernthema der Arbeit: Die Entwicklung einer Hardwareplattform, mit welcher die zeit- und kostenaufwendige, elektrische Kontaktierung von PCBs bei der Sicherheitsanalyse automatisiert werden kann. Nach Festlegung des Designs soll diese Plattform in einem ersten, mechanischen Prototypen umgesetzt werden. Es handelt sich daher um eine interdisziplinäre Arbeit, welche die Fachgebiete der Informatik, des Maschinenbaus und bedingt auch der Elektrotechnik umspannt.

1.3. Beiträge und Aufbau der Arbeit

Die Arbeit lässt sich in fünf Themengebiete unterteilen. Diese werden nachfolgend zusammen mit der Struktur dieser Arbeit kurz vorgestellt.

Visuelle PCB-Analyse

Zunächst wurden neue Verfahren zur visuellen PCB-Analyse untersucht. Dabei wurde der Fokus auf Neuronale Netze zur Bauteil- und Pinerkennung gelegt. Mit Neuronalen Netzen konnten in den vergangenen Jahren große Fortschritte im Bereich der Bildverarbeitung und Objekterkennung erzielt werden [4]. Auf einem realitätsnahen Bilddatensatz mit PCB-Aufnahmen konnten mit diesen Modellen ebenfalls sehr gute Ergebnisse bei der Erkennung einzelner Bausteine auf bestückten PCBs erzielt werden. Kapitel 2 behandelt die ausgewählten Modelle und deren Training. Ferner wird in diesem Kapitel eine entwickelte, klassische Computer Vision Pipeline mit Neuronalen Netzen zur Lokalisierung von Komponentenpins verglichen. Mit beiden Verfahren konnten auf einem Beispieldatensatz gute Ergebnisse erzielt werden. Zur Evaluation der untersuchten Verfahren wurden jeweils dem Stand der Technik entsprechende, neu zusammengestellte Bilddatensätze sowie etablierte Metriken zur Bewertung der Detektionsleistung genutzt.

Hardwareplattform zur automatisierten PCB-Kontaktierung

Kapitel 3 widmet sich der entwickelten Hardwareplattform und der Umsetzung in Form eines realen Prototyps. Dazu wurden zunächst die Anforderungen an ein solches System definiert und mögliche, bereits existierende Plattformen verglichen. Die Wahl fiel letztlich auf eine Neukonstruktion. Basierend auf den Anforderungen, wurde hierfür zunächst eine geeignete Kinematik ausgewählt und Entscheidungen zur Auswahl wichtiger, mechanischer Komponenten getroffen. Der finale Entwurf wurde in Form eines realen Prototyps umgesetzt. Dieser zeigte in ersten Tests gute Ergebnisse bezüglich der absoluten Positionier- und Wiederholgenauigkeit.

Kalibrierung der Hardwareplattform

Die Genauigkeit der entwickelten Kontaktierhardware wird primär durch ihre initiale Kalibrierung bestimmt. Hierfür wurde eine automatisierte Routine entwickelt. Durch diese ist es möglich, Fertigungs- und Montagetoleranzen zu kompensieren. In Kapitel 4 wird dieses Verfahren, zusammen mit den finalen Evaluationsergebnissen, der entwickelten Hardwareplattform vorgestellt.

Softwareprototyp zur Interaktion mit der Kontaktierhardware

Zur Ansteuerung der Hardwareplattform wurde eine prototypische Software mit einer grafischen Bedienoberfläche implementiert. Mit dieser ist es möglich, die aufgebaute Kontaktierhardware zu kalibrieren und einfache, teilautomatisierte Tests durchzuführen. Ein solcher Test, der auch im Zuge der Evaluation der Plattform durchgeführt wurde, ist das interaktive Anfahren von Testpunkten. Dabei wird durch den Benutzer ein Punkt im Kamerabild selektiert, woraufhin das System eine zuvor ausgewählte Prüfspitze an dieser Stelle platziert. Eine kurze Beschreibung der Software findet sich zusammen mit den Ergebnissen der durchgeföhrten Evaluation in Kapitel 4.

Erzeugung hochaufgelöster PCB-Aufnahmen

Zur Analyse kleiner Strukturen auf einem PCB werden hochaufgelöste Bildaufnahmen benötigt. Ein Verfahren, das zur Generierung solcher Aufnahmen mit dem Kamerasystem der aufgebauten Hardwareplattform eingesetzt werden kann, wird ebenfalls am Ende von Kapitel 4 beschrieben.

Die vorliegende Arbeit behandelt Aufgabenstellungen aus sehr unterschiedlichen Themenbereichen. Zur besseren Lesbarkeit finden sich daher die notwendigen theoretischen Grundlagen sowie verwandte wissenschaftliche Arbeiten zu Beginn der jeweiligen Teilkapitel. Auch Anregungen für Weiterentwicklung in zukünftigen Arbeiten werden jeweils am Ende der einzelnen Kapitel gegeben.

2. Visuelle PCB-Analyse

Nachdem eine zu untersuchende Platine durch Entfernung des Gehäuses freigelegt wurde, erfolgt bei der manuellen PCB-Sicherheitsanalyse, wie auch bei einem Reverse Engineering, im ersten Schritt eine visuelle Inspektion. Das Ziel hierbei ist es, einen Überblick über die Funktionsweise des PCBs, die darauf verbauten Komponenten, deren Anordnung und Verbindungen untereinander zu erlangen. Bei der Sicherheitsanalyse wird dabei der Fokus primär auf aktive Bauteile, wie etwa Mikrocontroller oder Speicherchips gelegt. Passive Bauelemente, wie Widerstände oder Kondensatoren, sind zwar für das Gesamtverständnis wichtig, aber für die weitere Sicherheitsanalyse meist nicht weiter von Belang¹. Aufgrund dieser Tatsache wird im Weiteren der Begriff *Komponente* synonym für aktive Bauelemente verwendet.

Wurden potentiell sicherheitsrelevante Komponenten lokalisiert, werden im nächsten Schritt detailliertere Informationen zu diesen recherchiert. Ausgangspunkt für die Recherche sind meist aufgedruckte Komponenten-IDs der Hersteller, anhand derer ein Chip eindeutig identifiziert werden kann. Ein weiteres Merkmal zur Identifizierung ist außerdem das Chipgehäuse, das auch als *Package* bezeichnet wird. Dieses zeichnet sich insbesondere durch die Art der Montage des Chips, wie etwa Through Hole Technology (THT), Surface-Mounted Device (SMD) oder Ball Grid Array (BGA), sowie die Anzahl der Komponentenpins aus [5, S. 15 ff.].

Eine wichtige Informationsquelle sind häufig Datenblätter der Chiphersteller. Falls diese öffentlich zugänglich sind, können aus ihnen wichtige Informationen, wie etwa die Pinbelegungstabelle entnommen werden. Diese definiert die Zuordnung einzelner Funktionen zu den Pins einer Komponente. Zusammen mit den ebenfalls in den Datenblättern dokumentierten Informationen über vorhandene Funktionalitäten, können so leicht mögliche Angriffspunkte einer Komponente identifiziert werden. Hierzu zählen insbesondere exponierte Debugging-Schnittstellen und Daten- bzw. Kommunikationsbusse, da diese eine Interaktion und den Datenaustausch ermöglichen.

Im Rahmen dieser Arbeit wurden neue Verfahren für die Automatisierung von zwei grundlegenden Problemstellungen der visuellen PCB-Analyse untersucht: Zum einen die Lokalisierung einzelner Komponenten auf einem bestückten PCB und zum anderen die möglichst exakte Bestimmung der Positionen einzelner Bauteilpins. Letzteres ist Ausgangspunkt für den nächsten Schritt einer automatisierten Sicherheitsanalyse - die elektrische Pinkontaktierung. Diese erlaubt die Interaktion mit einzelnen Bausteinen über elektrische Signale oder auch die Rekonstruktion der Leiterbahnen in Form einer Netzliste, um ein tieferes Verständnis einer Platine zu erlangen.

Zunächst werden in Kapitel 2.1 aktuell eingesetzte Verfahren zur visuellen PCB-Analyse vorge-

¹Ergebnis eines Interviews mit Sicherheitsanalysten der SCHUTZWERK GmbH am 05.03.2019

stellt. Da es sich bei der Chiperkennung im Wesentlichen um eine Aufgabe der Objekterkennung handelt, werden an dieser Stelle auch aktuell eingesetzte Techniken für diese Problemstellung erwähnt. Insbesondere wird dabei auf den Einsatz Neuronaler Netze eingegangen, durch die in den letzten Jahren große Fortschritte auf diesem Gebiet erzielt werden konnten. Im nachfolgenden Kapitel 2.2 werden die Grundlagen Neuronaler Netze zur Objekterkennung und wichtige, im Rahmen dieser Arbeit eingesetzte Methoden der Bildverarbeitung kurz umrissen. In Kapitel 2.3 wird dann ein entwickeltes Verfahren zur Chiperkennung mittels Neuronaler Netze vorgestellt. Es wird auf die verwendeten Bilddatensätze zum Training und die untersuchten Netzwerktopologien eingegangen. Die auf einem Evaluationsdatensatz erzielten Ergebnisse werden vorgestellt und mit denen existierender Arbeiten verglichen. Gegenstand von Kapitel 2.4 sind neue Lösungsansätze zur möglichst exakten Lokalisierung von Komponentenpins. Hierbei wurde ebenfalls der Einsatz Neuronaler Netze erprobt und mit einer klassischen Computer Vision Pipeline verglichen. Das Kapitel schließt mit einer Zusammenfassung der erzielten Ergebnisse in den jeweiligen Teilbereichen der visuellen PCB-Analyse und gibt Anregungen für weitere Entwicklungen basierend auf den gewonnenen Erkenntnissen (Kapitel 2.5).

2.1. Verwandte Arbeiten

Bei Verfahren zur Analyse von PCBs muss grundsätzlich zwischen zwei Kategorien differenziert werden: Verfahren zur Analyse von unbestückten PCBs beschäftigten sich mit der Untersuchung von unbestückten Platinen, bei denen noch keine Bausteine aufgelötet oder diese im Zuge einer Analyse oder Reparatur wieder entfernt wurden. In diesem Fall sind lediglich die kupfer- oder goldfarbenen Leiterbahnen und Lötpads, der farbige Lötstopplack und eventuell ein aufgebrachter *Silkscreen* auf den PCBs vorhanden. Im Gegensatz dazu existieren Verfahren, die sich mit der Analyse von vollständig bestückten PCBs beschäftigen.

Visuelle Analyse unbestückter Platinen

Bei der Analyse von unbestückten Platinen kommen häufig auf Farbinformationen basierende Segmentierungsverfahren zum Einsatz. Johnson stellt in seiner Arbeit einen Ansatz zur Rekonstruktion von Netzlisten auf Basis von PCB-Bildern vor [6]. Das beschriebene Verfahren ist in der Lage, die Netzliste von ein- und zweilagigen, unbestückten PCBs ohne Lötstopplack und *Silkscreen* anhand von Bildern der Ober- und Unterseite der Platine zu generieren. Zunächst werden die Bilder mittels Graustufenkonvertierung und Schwellwertbildung binarisiert. Anschließend werden die Netze, d.h. die Kupferbahnen, welche einzelne Bausteine elektrisch miteinander verbinden, für jede Platinenseite individuell mittels Konturfindung bestimmt. Erkannte Löcher in den Konturen werden dabei als Verbindungen von Ober- und Unterseite interpretiert. Anschließend erfolgt eine automatisierte Komponentenerkennung über ein Template-Matching mit hinterlegten und entsprechend skalierten Komponenten-Footprints. Die finale Netzliste wird abschließend aus den gewonnenen Informationen über die Zuordnung der Komponenten zu

den Pads und damit den entsprechenden Netzen generiert. Dabei werden die Ergebnisse der Ober- und Unterseite über die im ersten Schritt ermittelten Verbindungsbohrungen miteinander kombiniert. Das Verfahren konnte auf der von Johnson verwendeten Testplatine alle Netze und Komponenten korrekt identifizieren. Die Zahl der zusätzlich, fälschlicherweise erkannten Komponenten war sehr gering. Aufgrund eines Implementierungsfehlers schlug die Verknüpfung der Ober- und Unterseite fehl. Die Komponentenerkennung erwies sich als langsam und unflexibel. Als mögliche Verbesserung schlägt Johnson eine Eingrenzung der untersuchten Regionen mittels Heuristiken vor, beispielsweise das Ignorieren von leeren Flächen.

Auch Tan und Kit nutzen Farbinformationen in einem zweistufigen Clusteringverfahren zur Analyse von unbestückten PCBs [7]. Ziel des Verfahrens ist die möglichst exakte Positionsbestimmung der Testpads in Bildern von unbestückten PCBs. Als Testpads werden speziell auf einem PCB angebrachte Kontaktstellen bezeichnet, die eine elektrische Kontaktierung während des Platinentests in der Herstellung ermöglichen [5, S. 137]. Tan und Kit setzen im ersten Schritt den K-Means Clustering Algorithmus ein, um Pixel entsprechend ihrer Farbe zu gruppieren. Ziel hierbei ist das Finden von Clustern mit grauen Pixeln, von denen angenommen wird, dass es sich hierbei um Testpads handelt. Auf die so erhaltenen Ergebnisse wird im nächsten Schritt ein weiteres Clustering auf den Bildkoordinaten durchgeführt. Tan und Kit evaluieren hierfür sowohl den K-Means als auch den Two-Step Algorithmus, wobei sich der Two-Step Algorithmus als robuster gegenüber sporadisch auftretendem Pixelrauschen erwies. Auf dem zur Evaluierung verwendeten Testbild konnte mit diesem Verfahren eine *Precision* von 93,75 % bei einem *Recall* von 100 % erzielt werden.

Mata et al. nutzen in ihrer Arbeit Farbunterschiede zwischen den Leiterbahnen und dem Lötstopplack, um Leiterbahnkonturen unbestückter Platten zu extrahieren [8]. In dem von ihnen vorgestellten Verfahren wird das Platinenbild zunächst in ein Graustufenbild konvertiert. Anschließend wird eine Kantenerkennung durchgeführt. Nach einer Vektorisierung verwenden Mata et al. die gewonnenen Leiterbahnkonturen als Negativ zum Ätzen von Replikaten der ursprünglichen PCBs.

Visuelle Analyse bestückter Platten

Im Gegensatz zu den bisher genannten Arbeiten beschäftigen sich Kleber et al. in ihrer Arbeit mit der Analyse von vollbestückten PCBs [9]. Sie nutzen ein auf Farbsegmentierung basierendes Verfahren, um einzelne Chips auf Platten im Zuge des *Reverse Engineering* zu lokalisieren. Nach einer Vorverarbeitung des Bildmaterials durch morphologische Operationen und einer Transformation in den **Hue Ssaturation Value** (HSV)-Farbraum nutzen sie ein Schwellwertverfahren, um dunkle Bereiche aus dem Gesamtbild herauszulösen. Auf die verbliebenen Bildbereiche wenden sie den *Watershed*-Algorithmus zur Segmentierung der einzelnen Chips an. Ein abschließendes Rechteck-Fitting liefert die Bounding Boxen der erkannten Chips. Mit diesem Verfahren konnten sie eine Komponentenerkennungsrate von über 65 % auf ihrem Testdatensatz erzielen. Für weiterführende Arbeiten zur Erkennung einzelner Bausteinpins erwähnen Kleber et

al. den Einsatz der von ihnen untersuchten Methoden zur Objekterkennung, namentlich die Farbsegmentierung und das Template bzw. Feature basierte Matching unter Einbeziehung der erkannten Chip Bounding Boxen.

Die Chiperkennung auf bestückten PCBs lässt sich generell der Problemkategorie der Objekterkennung zuordnen. In diesem Forschungsgebiet konnten in den vergangenen Jahren große Fortschritte durch den Einsatz von Neuronalen Netzen und insbesondere durch sogenannte *Convolutional Neural Networks* (CNNs) erzielt werden [4]. Die Objektdetektionsleistung und Ausführungsgeschwindigkeit dieser CNNs hängt dabei sehr stark von der gewählten Netztopologie ab. In regionsbasierten, zweistufigen Netzen, wie das von Girshick et al. entwickelte *R-CNN* [10] und den Weiterentwicklungen *Fast R-CNN* [11] und *Faster R-CNN* [12], werden in einem ersten Schritt sogenannte *Regions of Interest* (ROIs) geschätzt. Bei ROIs handelt es sich um Bildbereiche, die potentiell zu detektierende Objekte enthalten können. Die Klassifikation und finale Bestimmung der Bounding Boxen der einzelnen Objekte wird anschließend von einem oder mehreren CNNs durchgeführt. Im Gegensatz dazu erfolgt in sogenannten *Single Shot Detektoren* wie *YOLO* [13] oder auch *SSD* [14] die gesamte Objektdetektion in einem einzigen Modelldurchlauf.

Für das Training Neuronaler Netze werden Bilddatensätze benötigt, in denen die zu erkennenden Objekte manuell markiert und eindeutig einer Objektkategorie zugeordnet wurden. Einen solchen Datensatz für PCBs stellen Pramerdorfer und Kampel in ihrer Arbeit vor [15]. Der als *PCB-DSLR* betitelte Datensatz wurde primär für die automatisierte visuelle Analyse im Anwendungsbereich des Recyclings von PCBs zusammengestellt. Er besteht aus 748 Bildern von Platten mit insgesamt 9313 einzeln gelabelten Integrated Circuits (ICs). Die PCBs wurden mit einer professionellen DSLR Kamera in einer gut ausgeleuchteten Szene auf schwarzem Hintergrund aufgenommen. Neben den gelabelten ICs enthält der Datensatz weiterhin pixelgenaue Masken für die jeweils abgebildeten PCBs.

2.2. Grundlagen

In diesem Kapitel werden wichtige Grundlagen der im Weiteren eingesetzten Verfahren zur Objekterkennung und Bildsegmentierung beschrieben. Aufgrund des Umfangs der einzelnen Themen können an dieser Stelle lediglich die für das grundlegende Verständnis wichtigen Kernpunkte dargestellt werden. Kapitel 2.2.1 gibt einen Einblick in den Aufbau und die Funktionsweise Neuronaler Netze im Anwendungskontext der Objekterkennung. Dabei wird auch auf den Trainingsprozess dieser Modelle und die dafür benötigten Trainingsdaten eingegangen. Eine etablierte Metrik zur Beurteilung von Objektdetektoren, die bei der *Pascal VOC Challenge* [16] zum Einsatz kommt, wird in Kapitel 2.2.2 vorgestellt. Die restlichen Kapitel widmen sich Verfahren der klassischen Bildverarbeitung. Es wird auf die Segmentierung mittels Schwellwertbildung (Kapitel 2.2.3), morphologische Operationen zur Bildsegmentierung und Bereinigung binarisierter Bilder (Kapitel 2.2.4) und den *Floodfill*-Algorithmus (Kapitel 2.2.5)

eingegangen. Weiterhin werden der bilaterale Filter zur Vorverarbeitung und Glättung von Bildaufnahmen in Kapitel 2.2.6, die *Distance Transform* zum Ausdünnen und *Skeletieren* von flächigen Bereichen (Kapitel 2.2.7) und die Bestimmung von Zusammenhangskomponenten vorgestellt (Kapitel 2.2.8).

2.2.1. Neuronale Netze zur Objekterkennung

In diesem Abschnitt soll ein kurzer Überblick über die grundlegenden Mechanismen technischer Neuronaler Netze zur Objekterkennung gegeben werden. Aufgrund der Komplexität des Themas werden nur die für das grundlegende Verständnis notwendigen Begriffe und Prinzipien erklärt. Für Details sei an dieser Stelle auf die jeweils genannten Literaturquellen verwiesen.

Durch den Einsatz sogenannter *Convolutional Neural Networks*, eine spezielle Architektur Neuronaler Netze, konnten in den letzten Jahren große Fortschritte im Bereich der Objekterkennung erzielt werden [4][17]. CNNs sind in Schichten aufgebaut, wie Abbildung 2.1 zeigt. Dabei kommen primär drei unterschiedliche Schichttypen zum Einsatz.

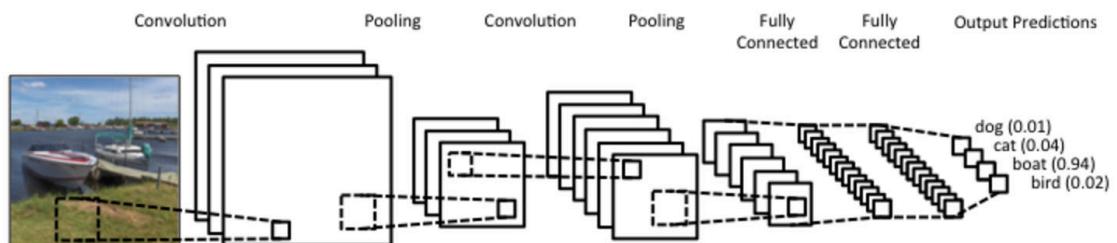


Abbildung 2.1.: Prinzipieller Aufbau eines CNNs [18]

Convolutional Layer

Die Basis bilden sogenannte *Convolutional Layer*, welche dieser Art von Neuronalen Netzen auch ihren Namen verleiht. In diesen Schichten werden aus der klassischen Bildverarbeitung bekannte Faltungsoperationen eingesetzt, um Merkmale zu extrahieren. Dazu wird ein Filterkern in Form einer Matrix über das Eingabebild oder eine bereits extrahierte Feature-Map bewegt. An jeder Position werden die Ausgabedaten durch Multiplikation der Daten des Eingabebildes an dieser Stelle mit den Gewichten des Filterkerne und anschließender Aufsummierung berechnet. Die eingesetzten Filterkerne können an dieser Stelle als Feature-Detektoren verstanden werden. Bei den detektierten Features kann es sich beispielsweise um Kanten oder Farbübergänge handeln.

Pooling Layer

Anschließend erfolgt häufig eine Größenreduktion der so erzeugten Feature-Maps mittels sogenannter *Pooling Layer*. Diese bestimmen die lokalen Maxima in einem Fenster definierter

Größe innerhalb der Feature-Maps. Ziel hierbei ist die Objekterkennung robuster gegenüber kleinen Positionsunterschieden der Features zu machen. Durch Verkettung mehrerer Blöcke aus *Convolutional*- und *Pooling*-Layer können ganze Merkmalshierarchien aufgebaut werden. Die von Simonyan und Zisserman entwickelte *VGG16*-Architektur [19] besitzt beispielsweise 13 *Convolutional* und 5 *Pooling Layer* zur Feature-Extraktion.

Fully Connected Layer

Die finale Objektklassifikation und Bestimmung der Bounding Box-Koordinaten wird mittels sogenannter *Fully Connected Layer* durchgeführt. Diese sind als Schicht aus mehreren technischen Neuronen zu verstehen. Jedes Neuron erhält dabei alle Ausgangsdaten der vorherigen Schicht als Eingangsdaten. Jeder Eingangsparameter wird in einem Neuron mit einem separaten Gewichtsfaktor multipliziert, aufsummiert und ein Offsetparameter addiert. Überschreitet der so berechnete Wert ein bestimmtes Niveau, erzeugt das Neuron entsprechend einer Transferfunktion einen Ausgabewert. Die Anzahl der technischen Neuronen eines *Fully Connected*-Layers ist variabel und bestimmt die Anzahl der Ausgabewerte. Häufig werden mehrere *Fully Connected Layer* am Ende eines CNNs hintereinandergeschaltet. Das genannte *VGG16*-Modell sieht etwa drei solcher Layer als Klassifikator vor.

Training Neuronaler Netze

Die Elemente der Filterkerne in den *Convolutional Layern* sowie die Gewichtungsfaktoren und Offsetwerte in den *Fully Connected Layern* sind zunächst unbekannt. Sie werden im Zuge des Modelltrainings ermittelt. Für das Training eines CNNs zur Objekterkennung werden Bilder benötigt, in denen die sichtbaren Objekte zuvor manuell gelabelt wurden. Unter dem Begriff des *Labelings* wird hier die Bestimmung der Bounding Box-Koordinaten und die eindeutige Zuordnung der sichtbaren Objekte zu einer Objektkategorie verstanden. Die so entstandenen Daten werden oft auch als *Ground Truth*-Daten bezeichnet.

Im Zuge des Trainings wird versucht, die Modellparameter so einzustellen, dass die Abweichungen zwischen den gelabelten Trainingsdaten und dem durch das Neuronale Netz erzeugten Output minimal werden. Es handelt sich dabei also um ein Optimierungsproblem. Zur Lösung dieses Problems werden überwiegend gradientenbasierte Optimierungsverfahren eingesetzt. Die Bilder des Trainingsdatensatzes durchlaufen dazu das Neuronale Netz. Anschließend wird der Modellfehler durch Vergleich der gelabelten Daten mit der Ausgabe des Modells durch Berechnung einer Fehlerfunktion bestimmt. Im nächsten Schritt wird der Gradient der Fehlerfunktion bezüglich jedes einzelnen Parameters des Modells berechnet. Mit diesen Daten werden die Modellparameter durch ein Optimierungsverfahren für den nächsten Iterationsschritt des Trainings angepasst. Dieser Ablauf wird solange wiederholt, bis ein definiertes Abbruchkriterium, etwa eine minimale Änderung des Modellfehlers, erfüllt wird. Neben dem einfachen Gradientenabstiegsverfahren, bei dem ein Parameter im Zuge einer Iteration des Optimierungsprozesses in negativer Gradientenrichtung um den Betrag des Gradienten multipliziert mit einer fixen

Lernrate verändert wird, kommen auch beschleunigende Verfahren, wie etwa das klassische Momentum-Verfahren [20], zum Einsatz. Zur Beschleunigung des Trainings wird oftmals bereits nach einer Teilmenge der Trainingsbilder eine Anpassung der Modellparameter durchgeführt. Die Anzahl der Bilder in dieser Teilmenge wird als *Batch size* bezeichnet.

Frameworks und Modelle

Für die Implementierung Neuronaler Netze stehen mehrere etablierte Frameworks zur Verfügung. Das von Google entwickelte Framework *Tensorflow* [21] ist eines davon. Neuronale Netze werden darin als gerichteter Graph elementarer Berechnungsoperationen, wie Addition oder Multiplikation, von Vektoren beschrieben. Durch Kenntnis dieses Berechnungsgraphen ist das Framework in der Lage, die für das Training Neuronaler Netze notwendige Gradientenbestimmung vollautomatisch und sehr effizient durchzuführen. Weiterhin werden für das Training Implementierungen gängiger Optimierungsverfahren, wie das einfache Gradientenabstiegsverfahren oder das Momentum-Optimierungsverfahren, zur Verfügung gestellt.

Basierend auf diesem Framework existieren weitere Bibliotheken, welche die Funktionalität Neuronaler Netze auf einer höheren Abstraktionsebene für spezifische Anwendungsbereiche zugänglich machen. Ein speziell für die Objekterkennung entwickeltes Projekt ist die auf Github veröffentlichte *Tensorflow Object Detection API* [22]. Das Projekt ermöglicht die Konfiguration einer kompletten Pipeline zum Training eines Neuronalen Netzes durch eine einzige Konfigurationsdatei. Neben der Definition des zu trainierenden Neuronalen Netzes enthält eine solche Trainingspipeline auch eventuell notwendige Vorverarbeitungsschritte der Trainingsbilder, wie beispielsweise die Skalierung der Bilder auf eine feste Größe. Auch Parameter für das Training des Neuronalen Netzes werden in der Pipeline konfiguriert. Als zentrale Parameter sind hierbei das gewählte Optimierungsverfahren und auch die Anpassung der Lernrate während des Trainings zu nennen.

Das Projekt stellt weiterhin bereits auf gängigen Datensätzen, wie etwa dem *Microsoft COCO (Common Objects in Context)* Datensatz [23], trainierte Modelle etablierter Neuronaler Netze für die Objekterkennung zur Verfügung. Enthalten sind sowohl auf Geschwindigkeit optimierte *Single Shot Detektoren*, wie das *SSD* [14] Modell, aber auch komplexere und leistungsfähigere Modelle, wie etwa eine Reimplementierung des *Faster R-CNN* Modells [12].

Diese vortrainierten Modelle können als Ausgangspunkt für das Training von Objektdetektoren neuer Objektklassen verwendet werden. Der neue, zu trainierende Objektdetektor profitiert dabei von den bereits gelernten Modellparametern in den niedrigeren Schichten des Modells. Insbesondere beim Training auf kleinen Datensätzen kann der Einsatz von *Transferlearning* zu besseren Ergebnissen führen [24].

2.2.2. Die Metrik der *Pascal VOC Challenge*

Es gibt eine Vielzahl unterschiedlicher Metriken zur Bewertung von Objektdetektoren. Bekannte Wettbewerbe für Aufgaben der Objekterkennung und Bildsegmentierung, wie die *Pascal VOC Challenge* [16] oder die *COCO Detection Challenge* [23], nutzen oft eigens definierte Metriken zur Bewertung der Kandidaten. In dieser Arbeit wird zur Beurteilung der untersuchten Verfahren der Chiperkennung die Metrik der *Pascal VOC Challenge* herangezogen. Durch die Bewertung anhand einer Kennzahl, der sogenannten *mean Average Precision*, und einem Graphen ist diese Metrik leicht verständlich und erlaubt einen einfachen Vergleich unterschiedlicher Verfahren.

Zu Berechnung der Metrik werden detektierte Objekte in vier Kategorien eingeteilt:

- **True Positives (TP)** bezeichnen korrekt erkannte Objekte
- Als **False Positives (FP)** werden Detektionen verstanden, die nicht in den *Ground Truth*-Daten vorhanden sind
- Nicht erkannte Objekte werden als **False Negatives (FN)** bezeichnet

Die vierte Kategorie der **True Negatives (TN)** beschreibt korrekt erkannte, nicht vorhandene Objekte. Diese Kategorie wird in der *Pascal VOC Challenge* jedoch nicht verwendet. Dies ist darin begründet, dass in einem Bild eine beliebig große Anzahl an Bounding Boxen für nicht vorhandene Objekte definiert werden kann.

Ob ein Objekt richtig erkannt wurde, wird durch einen Schwellwert für den Kennwert der Intersection over Union (IoU) bestimmt. Die Kennzahl IoU beschreibt dabei das Verhältnis der Schnittmenge zweier Bounding Boxen zur vereinigten Menge dieser Boxen. Nur wenn der IoU-Wert für eine Detektion über dem definierten Schwellwert liegt, gilt eine Detektion als korrekt erkannt.

Die Beurteilung eines Objektdetektors bei der *Pascal VOC Challenge* basiert im Wesentlichen auf zwei Kennzahlen: Die *Precision* ist als das Verhältnis der in die Kategorie TP fallenden Detektionen und der Gesamtzahl der Detektionen des Objektdetektors definiert.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Die zweite wichtige Kennzahl wird als *Recall* bezeichnet. Sie beschreibt das Verhältnis der TP-Detektionen zu den insgesamt in den *Ground Truth*-Daten vorkommenden Objekten. Letzteres lässt sich auch als Summe der TP- und FN-Detektionen ausdrücken. Der *Recall* wird dann wie folgt bestimmt

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Ein hoher Wert der *Precision* bedeutet, dass ein Objektdetektor nur Detektionsergebnisse an Stellen liefert, an denen auch wirklich Objekte sichtbar sind. Ein hoher *Recall*-Wert zeigt an, dass von einem Objektdetektor viele der sichtbaren Objekte erkannt werden. Für eine ganzheitliche

Beurteilung eines Objektdetektors werden die beiden Kennwerte in einem Diagramm aufgetragen. Der Wert der *Precision* wird dabei über den Werten des *Recalls* aufgetragen. Bei einem guten Objektdetektor bleibt der Wert für die *Precision* auch bei steigendem *Recall* auf einem konstant hohen Niveau.

Neben den genannten Graphen wird in der *Pascal VOC Challenge* für einen einfacheren Vergleich unterschiedlicher Objektdetektoren die Kennzahl der *Average Precision* (AP) genutzt. Diese beschreibt die approximierte Fläche unterhalb der *Precision-Recall-Kurve*. Können von einem Objektdetektor mehrere Objektklassen erkannt werden, wird zusätzlich die Kennzahl der *mean Average Precision* (mAP) als die über alle Objektklassen gemittelte AP bestimmt.

2.2.3. Bildsegmentierung mittels globaler Schwellwertbildung

Ein sehr einfaches Verfahren zur Segmentierung von Objekten in Bildaufnahmen ist die globale Schwellwertbildung, die auch als globales *Thresholding* bezeichnet wird. Ritter und Wilson formulieren das globale Thresholding wie folgt [25, S. 157 f.]: Sei $a \in \mathbb{R}^X$ ein Kamerabild, im Folgenden auch nur als Bild bezeichnet, und $[h, k]$ mit $h, k \in \mathbb{R}$ ein gegebener Schwellwertbereich. Dann lässt sich für jedes Pixel eines binären Ausgabebildes $b \in \{0, 1\}^X$ der Wert berechnen zu

$$b(x) = \begin{cases} 1, & \text{falls } h \leq a(x) \leq k \\ 0, & \text{sonst} \end{cases} \quad (2.1)$$

für alle $x \in X$. Durch dieses Verfahren entsteht eine binäre Maske, in der Pixel des Hintergrunds den Wert 0 und alle Objektpixel den Wert 1 besitzen. Ritter und Wilson unterscheiden weiterhin zwei Spezialfälle der Gleichung (2.1): Falls alle Pixel, die einen Wert größer einem bestimmten Schwellwert als Objektpixel betrachtet werden sollen, entsteht das Ausgabebild gemäß nachfolgender Gleichung

$$b(x) = \begin{cases} 1, & \text{falls } a(x) \geq k \\ 0, & \text{sonst} \end{cases}$$

Gehören Pixel mit einem Wert unterhalb einer definierten Schranke zu den gesuchten Objekten, kann das Ausgabebild entsprechend gebildet werden zu

$$b(x) = \begin{cases} 1, & \text{falls } a(x) \leq k \\ 0, & \text{sonst} \end{cases}$$

Bei diesem Verfahren wird vorausgesetzt, dass sich alle Pixel eines Bildes in zwei Kategorien einteilen lassen. Es lässt sich auch auf mehrkanalige Farbbilder erweitern, bei denen dann für jeden Farbkanal ein separater Schwellwert bzw. Schwellwertbereich definiert wird.

2.2.4. Morphologische Operationen

Morphologische Operationen werden in der Bildverarbeitung zur Filterung und Strukturanalyse eingesetzt [25, S. 205-211]. Die zugrundeliegende mathematische Morphologie wurde entwickelt, um boolesche Operationen in der Bildebene zu formulieren. Die Basis bilden die morphologischen Operationen der *Dilatation* und *Erosion*, welche sich durch die *Minkowski Addition* und *Minkowski Subtraktion* formulieren lassen. Seien $A \subset \mathbb{R}^n$ und $B \subset \mathbb{R}^n$ zwei beliebige Teilmengen, so ist die *Minkowski Addition* definiert als

$$A \times B = \{a + b : a \in A, b \in B\} \quad (2.2)$$

und die *Minkowski Subtraktion* durch

$$A/B = (A' \times B^*)' \quad (2.3)$$

mit $B^* = \{-b : b \in B\}$ und $A' = \{x \in \mathbb{R}^n : x \notin A\}$. Darin bezeichnet B^* die Spiegelung von B am Ursprung $0 = (0, 0, \dots, 0) \in \mathbb{R}^n$ und A' das Komplement von A . Mit Gleichung (2.2) und (2.3) formulieren Ritter und Wilson die Operation der *Dilatation* in einem binären Bild $A \in \{0, 1\}^X$ mit einem Strukturelement $B \in \{0, 1\}^X$ durch die folgende Gleichung

$$b(x) = \begin{cases} 1, & \text{falls } x \in X \cap (A \times B) \\ 0, & \text{sonst} \end{cases}$$

Die *Erosion* definieren sie weiterhin zu

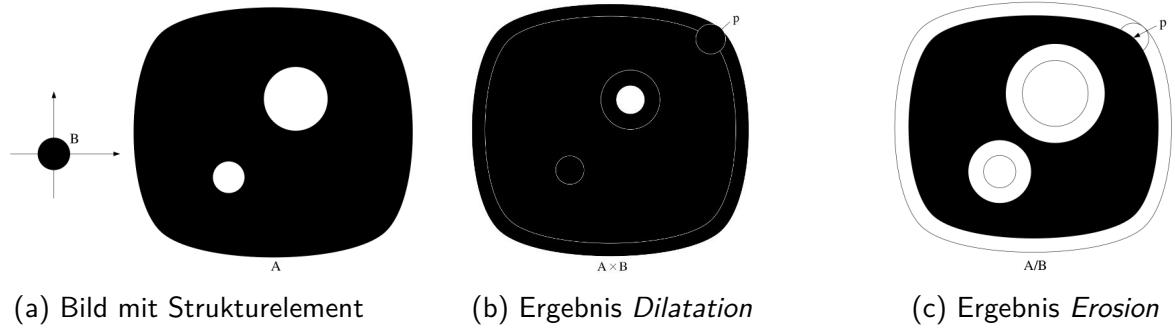
$$b(x) = \begin{cases} 1, & \text{falls } x \in A/B \\ 0, & \text{sonst} \end{cases}$$

Abbildung 2.2 stellt die beiden morphologischen Operationen auf einem Binärbild anhand eines Beispiels dar. Sowohl bei der *Dilatation* als auch *Erosion* wird das Strukturelement B über das Ausgangsbild A bewegt. An jeder Position p wird bei der *Dilatation* das Pixel des binären Ausgabebildes auf 1 gesetzt, falls eine Überschneidung von A und B vorliegt. Bei der morphologischen *Erosion* muss sich das Strukturelement B vollständig in A befinden, damit das entsprechende Pixel in der Ausgabemaske den Wert 1 erhält.

Dilatation kann genutzt werden, um kleine Löcher in Konturen zu schließen. Durch *Erosion* können kleine Konturen oder auch Brücken zwischen Teilkonturen entfernt werden. Eine kombinierte Anwendung von *Erosion* und anschließender Dilatation wird auch als *Opening* Operation bezeichnet und lässt sich formulieren als

$$A \circ B = (A/B) \times B$$

Die Anwendung der beiden Operationen in umgekehrter Reihenfolge wird *Closing* Operation genannt:

Abbildung 2.2.: Morphologische *Dilatation* und *Erosion* [25, S. 206 f.]

$$A \bullet B = (A \times B)/B$$

2.2.5. Floodfill

Torbert stellt in seiner Arbeit den als *floodfill* bezeichneten Algorithmus als Modellierungswerkzeug zur Verbreitung einer Krankheit vor [26]. Derselbe Algorithmus wird auch in der Computergrafik genutzt, um zusammenhängende Bildbereiche mit einheitlicher Farbe zu markieren [27]. Ausgehend von einem oder mehreren Ursprungspixeln, auch als *Seed-Pixel* bezeichnet, werden angrenzende Pixel auf farbliche Ähnlichkeit untersucht. Ein als ähnlich klassifiziertes Pixel wird anschließend in die Menge der zu dem *Seed-Pixel* gehörenden Komponente aufgenommen. Dieses Verfahren wird dann iterativ ausgehend von den neu hinzugekommenen Komponentenpixeln wiederholt.

Es gibt verschiedene Varianten dieses Algorithmus. Eine mögliche Variation betrifft die Nachbarschaftsbeziehung der Pixel. Es können beispielsweise Pixel in einer kreuzförmigen 4er-Nachbarschaft oder auch alle ein Pixel umgebende Nachbarpixel in der sogenannten 8er-Nachbarschaft für das *Floodfill* untersucht werden. Weiterhin wird zwischen Verfahren unterschieden, welche die farbliche Ähnlichkeit immer bezüglich der *Seed-Pixel* prüfen und solchen, die den Farbwert jeweils mit dem aktuellen Pixel durchführen. Letzteres erlaubt dem *Floodfill*-Algorithmus auch die Ausweitung über Farbgradienten hinweg.

2.2.6. Bilaterale Filterung

Der von Tomasi und Manduchi vorgestellte bilaterale Filter eignet sich zum Glätten von Bildern beim gleichzeitigen Erhalt von Kanteninformationen [28]. Es handelt sich dabei um einen nichtlinearen Filter. Der Wert eines gefilterten Pixels wird aus einer gewichteten Kombination der Nachbarpixel berechnet. Die Gewichtungsfunktion setzt sich dabei aus zwei Komponenten zusammen: Zum einen werden Nachbarpixel entsprechend ihrer räumlichen Nähe, etwa mit einer gaußschen Verteilungskurve, gewichtet. Zusätzlich werden diese aber auch entsprechend ihrer Ähnlichkeit im Hinblick auf die Farbe bzw. Intensität mit dem Zentrumspixel bewertet. Auch hier

kann beispielsweise eine gaußsche Verteilungskurve angesetzt werden. Die Gewichtungsfunktion des Filters setzt sich dann als Summe der beiden Einzelfaktoren zusammen.

2.2.7. Distance Transform

Ein Verfahren zur Bildkompression und des sogenannten *Skeletonizings*, bei dem dünne, drahtförmige Konturen aus flächigen Bereichen gewonnen werden, ist die sogenannte *Distance Transform* [25, S. 178 ff.]. Dabei wird jedem Vordergrundpixel einer binären Maske die Distanz zum nächstgelegenen Hintergrundpixel als Wert in einem Ausgabebild zugewiesen. Die finale Kontur des *Skeletons* kann durch Bestimmung lokaler Maxima gefunden werden.

Es können verschiedene Distanzmaße zur Berechnung der *Distance Transform* angesetzt werden. In der *OpenCV* Bibliothek [29] sind beispielsweise Implementierungen der L^1 - und L^2 -Vektornormen vorhanden. Die Länge $d \in \mathbb{R}$ zwischen zwei Punkten $p_1 = (x_1, y_1)$ und $p_2 = (x_2, y_2)$ mit $p_1, p_2 \in \mathbb{R}^2$ berechnet sich gemäß der L^1 -Vektornorm zu

$$d = |x_1 - x_2| + |y_1 - y_2|$$

2.2.8. Zusammenhangskomponenten

Das Ergebnis der Objektsegmentierung liefert in vielen Fällen eine binäre Maske $a \in \{0, 1\}^X$. In dieser sind potentiell interessante Bereiche durch Pixel mit dem Wert 1 markiert. Hintergrundpixel erhalten den Wert 0. Durch Algorithmen zur Bestimmung von Zusammenhangskomponenten werden zusammenhängenden Arealen individuelle Labels vergeben. Objekte können dann anhand ihres Labels angesprochen und gezielt weiter analysiert werden [25, S. 192].

Eine häufig durchgeführte Analyse auf Zusammenhangskomponenten ist die Konturfindung. Suzuki und Abe stellen in ihrer Arbeit Algorithmen zur topologischen Analyse in binären Bildern vor [30]. Die *OpenCV*-Bibliothek [29] enthält eine Implementierung des von Suzuki und Abe vorgestellten Algorithmus zur Konturfindung in binarisierten Bildern.

2.3. Chiperkennung mit Neuronalen Netzen

In diesem Kapitel wird ein neuer Ansatz zur Erkennung von Komponenten auf bestückten PCBs mittels Neuronaler Netze vorgestellt. In Kapitel 2.3.1 werden zunächst die ausgewählten und im weiteren Verlauf der Arbeit evaluierten Modelle beschrieben. Auf die Bilddatensätze, mit denen die Neuronalen Netze trainiert wurden, wird in Kapitel 2.3.2 näher eingegangen. Die für das Training genutzten Parameter sowie das dafür genutzte Rechnersystem wird in Kapitel 2.3.3 vorgestellt. Das Unterkapitel schließt mit einer Beschreibung der eingesetzten Evaluationsmethodik und Vorstellung der erzielten Ergebnisse bei der Chiperkennung in Kapitel 2.3.4.

2.3.1. Verwendete Modelle

Für die Chiperkennung im Rahmen dieser Arbeit wurden zwei Modelle aus dem *Tensorflow Object Detection API Model Zoo* untersucht: Zum einen wurde das dort vorkonfigurierte *Faster R-CNN* Modell ausgewählt. In der genutzten Implementierung wird, im Gegensatz zu der von Ren et al. ursprünglich vorgestellten Architektur [12], das *Inception Resnet v2* Modell [31] zur Extraktion von Features in den niedrigeren Layern verwendet. Wie auch das von Ren et al. ursprünglich vorgestellte Modell verarbeitet das Netz Bilder mit maximal 1024 Pixel in der größten Bilddimension. Dieses Modell wurde aufgrund seiner sehr guten Ergebnisse in dem *Microsoft COCO* Wettbewerb mit einer mAP von 37 % ausgesucht. Die theoretisch maximale Auflösung der Eingabebilder von 1024×1024 Pixeln sollte auch die Erkennung von kleineren Objekten ermöglichen, was bei Komponenten auf einem PCB oftmals der Fall ist. Aufgrund der Komplexität benötigt das Modell relativ viel Rechenleistung und Zeit für die Ausführung und das Training. Da bei einer vollautomatischen Sicherheitsanalyse der Schritt der visuellen Analyse im Regelfall nur ein Mal durchgeführt werden muss, spielt die Analysedauer an dieser Stelle nur eine untergeordnete Rolle.

Weiterhin wurde ein *Single Shot Detector* in Form eines *SSD* Modells trainiert. Es handelt sich dabei um die ebenfalls aus dem *Tensorflow Object Detection API Model Zoo* stammende Implementierung eines *SSD* Modells. Es nutzt zur Feature-Extraktion eine Implementierung des ResNets [32] mit 50 Layern. Im Gegensatz zu dem von Liu et al. entwickelten, ursprünglichen *SSD* Modell [14], erwartet das Modell die Bilddaten mit einer fixen Auflösung von 640×640 Pixeln. Mit 35 % mAP zeigt es eine ähnlich gute Leistung im *Microsoft COCO* Wettbewerb wie die Implementierung des *Faster R-CNN* Modells. Die Ausführungsgeschwindigkeit ist im Vergleich zu diesem Modell jedoch um etwa den Faktor 8 schneller und erlaubt daher auch eine visuelle PCB-Analyse auf weniger leistungsfähiger Computerhardware, wie etwa einem Raspberry Pi in angemessener Zeit.

2.3.2. Bilddatensätze

Die ausgewählten Neuronalen Netze wurden auf dem von Pramerdorfer und Kampel vorgestellten *PCB-DSLR* Datensatz [15] trainiert. Der schwarze Hintergrund wurde entfernt und der Bildausschnitt auf die Bounding Box der PCBs mit einem zusätzlichen Rand von 50 Pixeln in allen Richtungen beschränkt. Außerdem wurden die Rotationen der Bounding Boxen der gelabelten ICs entfernt, so dass zu den Achsen des Bildkoordinatensystems parallele Boxen entstanden. Dies entspricht auch der Ausgabe der verwendeten Neuronalen Netze. Die durchgeföhrten Transformationen sind in Abbildung 2.3 an einer Aufnahme beispielhaft dargestellt. Die gelabelten ICs sind darin als rote Rechtecke farblich hervorgehoben.

Von den insgesamt 748 Bildern des Datensatzes wurden zufällig 598 (ca. 80 %) für das Training ausgewählt. Die restlichen 149 Bilder wurden für die Evaluation und Beurteilung des Trainingsverlaufs zurückgehalten.

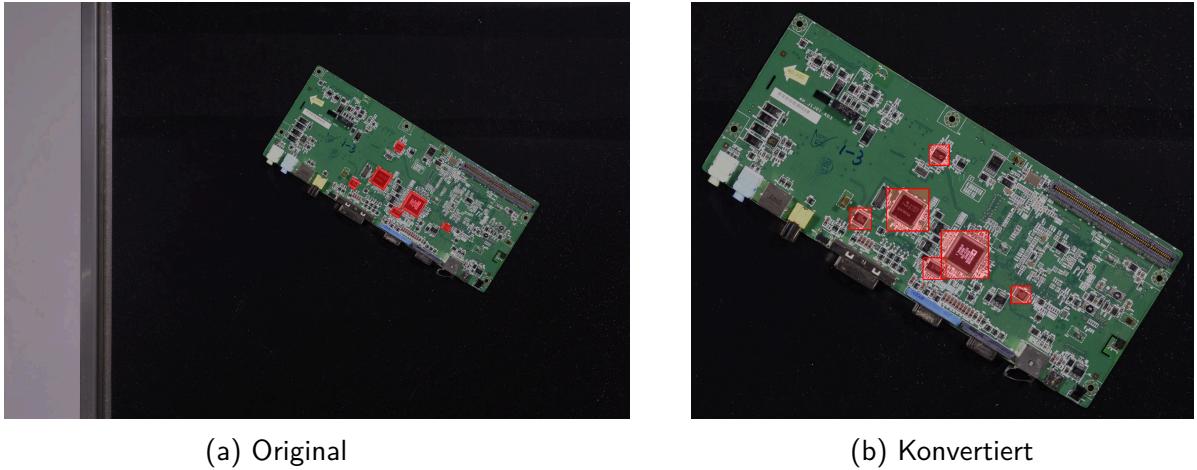


Abbildung 2.3.: Für das Training aufbereitete Aufnahme aus dem PCB-DSLR Datensatz [33] (PCB: 87, Recording 2); *Ground Truth*-Daten sind als rote Rechtecke eingezeichnet

Neben dem *PCB-DSLR* Datensatz wurde außerdem ein eigener Trainingsdatensatz zusammengestellt, der im Weiteren als *PCB-GOOGLE* Datensatz bezeichnet wird. Für die Zusammenstellung des Datensatzes wurden zunächst in der Suchmaschine Google² Bilder zu den Begriffen *microcontroller*, *pcb*, *pcb red* und *platine* gesucht. Die Bilder der Suchergebnisse wurden anschließend mit einem Skript automatisch heruntergeladen. Auf diese Weise konnte ein Rohdatensatz mit insgesamt 1223 Bildern generiert werden. Anschließend erfolgte eine manuelle Sortierung und Auswahl der Bilder, da der Rohdatensatz sehr viele nicht verwertbare Aufnahmen enthielt. Der finale Datensatz enthält mit 190 Bildern nur noch einen Bruchteil des Rohdatensatzes. Er weist eine große Varianz bezüglich der PCB- und Chipaufnahmen auf. So sind sowohl Aufnahmen von einzelnen, unverbauten Chips, überwiegend aber Bilder von bestückten PCBs in unterschiedlichen Farben enthalten. Die abgebildeten Chips variieren stark in ihrer Größe. Auch die Lichtverhältnisse der Aufnahmen schwanken stark. Eine Beispielaufnahme aus dem Datensatz für jedes der gesuchten Schlüsselworte in der Bildersuche zeigt Abbildung 2.4. Das Labeling der Chips erfolgte mit dem kostenlos online verfügbaren Labeling-Tool Labelbox³. Der finale Datensatz enthält 1100 einzeln gelabelte Chips, welche in Abbildung 2.4 ebenfalls als rote Rechtecke visualisiert sind. Dabei wurde nicht nach *Package* der Chips unterschieden. Alle Objekte besitzen die einheitliche Objektklasse *chip*.

Zur Evaluation der trainierten Modelle und der Bewertung des Nutzens für den praktischen Einsatz bei der Chiperkennung wurde ein weiterer, deutlich kleinerer Bilddatensatz aufgebaut. Dieser, nachfolgend als *PCB-CUSTOM* bezeichnete Datensatz, besteht aus Bildaufnahmen von Platinen, die bei der *SCHUTZWERK GmbH* selbst entworfen wurden, und weiteren, von den Sicherheitsanalysten der *SCHUTZWERK GmbH* als repräsentativ erachteten PCBs. Die Aufnahmen wurden mit einer Systemkamera und einem Teleobjektiv mit eingesetzten Zwischenringen

²<https://www.google.de/imghp>

³<https://labelbox.com/>

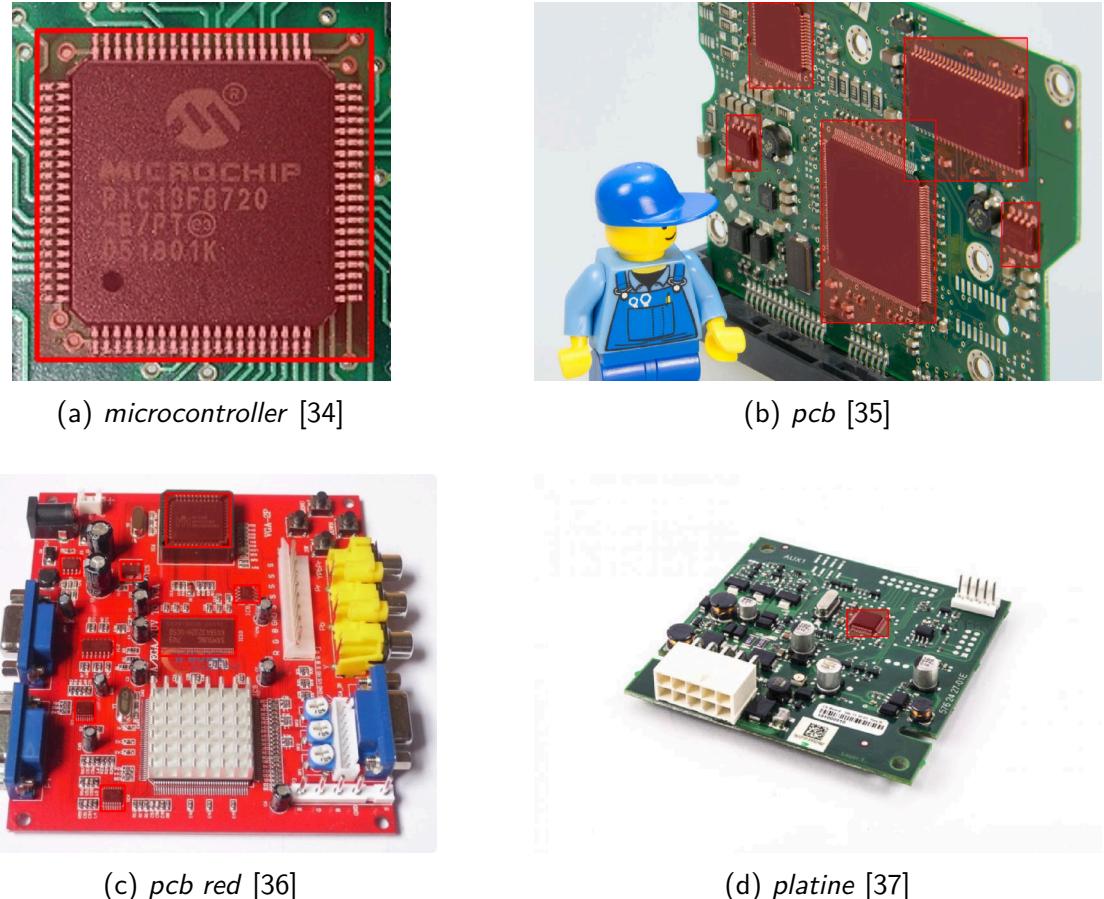


Abbildung 2.4.: Aufnahmen aus den verschiedenen Kategorien des selbst zusammengestellten *PCB-GOOGLE* Bilddatensatzes; *Ground Truth*-Daten sind als rote Rechtecke eingezeichnet

aufgenommen. Die so erzeugten Bilder besitzen eine Auflösung von 6000×4000 Pixeln. Der Datensatz besteht aus 15 Einzelaufnahmen, wobei besonders Wert auf möglichst unterschiedliche Farben des Lötstopplacks gelegt wurde, wie in Abbildung 2.5 zu sehen ist. Insbesondere auf Farbsegmentierung basierende Verfahren sind recht sensibel auf die Farbe der umliegenden Bereiche der zu erkennenden Objekte.

2.3.3. Training

Die vorgestellten Modelle wurden mit der *Tensorflow Object Detection API* [22] trainiert. Beide Neuronalen Netze wurden zum einen separat sowohl auf dem *PCB-DSLR*, als auch auf dem *PCB-GOOGLE* Datensatz trainiert. Außerdem erfolgte zusätzlich das Training des *Faster R-CNN* Modells auf einer Kombination beider Bilddatensätze. Als Ausgangspunkt hierbei diente das auf dem *PCB-DSLR* Datensatz trainierte Modell.

Für das Training wurde die Tensorflow-Implementierung des klassischen Momentum Optimierers verwendet. Sei w_t der aktuelle Wert eines Parameters $w \in \mathbb{R}$ zum Zeitpunkt t und $\nabla E(w_t) \in \mathbb{R}^x$ der Gradient der Fehlerfunktion bezüglich dem Parameter w , so wird der

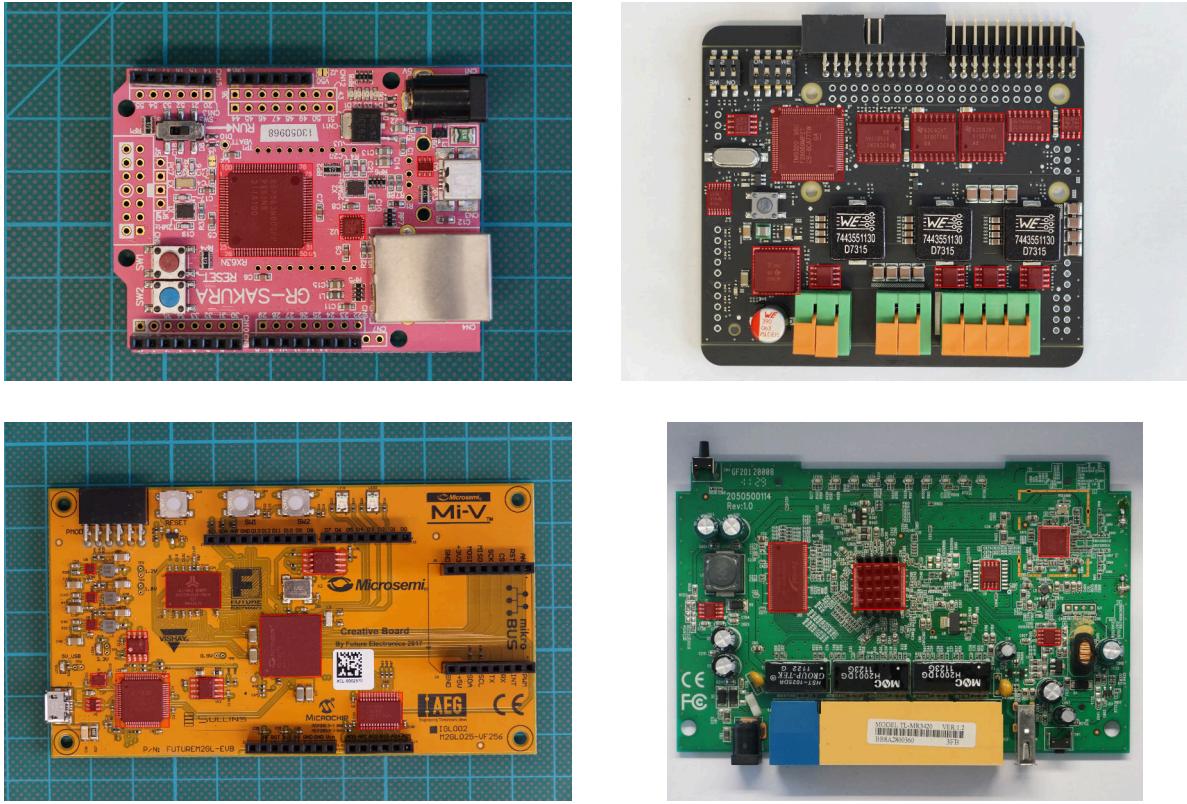


Abbildung 2.5.: Aufnahmen aus dem Bilddatensatz *PCB-CUSTOM*

Geschwindigkeitsterm v_t des Momentum-Verfahrens mit dem Momentum-Term $\beta \in \mathbb{R}$ wie folgt berechnet:

$$v_t = \beta * v_{t-1} + \nabla E(w_t)$$

Der Wert des Parameters im nächsten Iterationsschritt w_{t+1} berechnet sich dann mit der Lernrate $\alpha \in \mathbb{R}$ zu

$$w_{t+1} = w_t - \alpha * v_t$$

Der Momentum Term β wurde für beide Modelle bei dem Standardwert von 0,9 belassen. Für die Lernrate α wurde für das *Faster R-CNN* Modell ein nach Anzahl der trainierten Iterationen fest definierter Plan festgelegt. Die initiale Lernrate wurde bei dem Standardwert von $3 \cdot 10^{-4}$ belassen und nach 25.000 und 35.000 Iterationen um jeweils eine Zehnerpotenz verringert. Die Bestimmung dieser Schwellwerte erfolgte iterativ. Zunächst wurden die Modelle mit einer größeren Lernrate trainiert und der Fehler auf den Evaluationsdaten verfolgt. Sobald sich keine merklichen Änderungen des Fehlers mehr zeigten oder es zu periodischen Schwankungen kam, wurde das Training angehalten und mit einer reduzierten Lernrate fortgesetzt. Die Lernrate für das verwendete *SSD* Modell wurde mit dem in der Standardkonfiguration aus dem *Tensorflow Object Detection API Model Zoo* definierten, entsprechend einer Kosinusfunktion abnehmenden Lernrate trainiert [38]. Die Warmup-Lernrate betrug $1,3 \cdot 10^{-2}$ und wurde über 2000 Warmupiterationen auf die Basislernrate von $4 \cdot 10^{-2}$ erhöht. Anschließend wurde die

Lernrate entsprechend einer Kosinusfunktion über insgesamt 25.000 Iterationen verringert.

Für die Vorverarbeitung der Trainingsbilder stehen in der *Tensorflow Object Detection API* eine Vielzahl von Methoden zur Verfügung, um zusätzliche, künstliche Trainingsbilder zu generieren. Dies kann insbesondere bei kleinen Trainingsdatensätzen, wie den genutzten PCB-Bilddatensätzen, hilfreich sein, um trotzdem möglichst robuste Objektdetektoren zu erhalten. Die Trainingspipeline wurde zum einen so konfiguriert, dass die Trainingsbilder zufällig horizontal und vertikal gespiegelt werden. Auf diese Weise wurden künstlich neue Trainingsdaten generiert, die helfen sollen, die Modelle robuster gegenüber der Orientierung der aufgenommenen Chips zu machen. Weiterhin wurde in die Inputpipeline ein Verarbeitungsschritt aufgenommen, der ebenfalls zufällig nicht das gesamte Trainingsbild, sondern lediglich Bildausschnitte als Input an das zu trainierende Modell weiterleitet. Die Konfiguration erfolgte so, dass die erzeugten Bildausschnitte dabei zwischen 75 % und 100 % des ursprünglichen Bildes abdecken müssen und bezüglich der Seitenverhältnisse zwischen 0,75 und 3 des ursprünglichen Bildes variieren können.

Trainiert wurden die Modelle auf der von Google kostenlos zur Verfügung gestellten Plattform *Google Colab*⁴. Dort erhalten Benutzer eine virtuelle Maschine basierend auf der *Google Compute Engine*⁵, mit der über *Jupyter Notebooks*⁶ interagiert werden kann. Neben einer Umgebung, in der lediglich eine Intel(R) Xeon(R) vCPU mit 2,2 GHz Taktfrequenz und 12 GB RAM für Berechnungen zur Verfügung stehen, gibt es auch GPU und *Tensor Processing Unit* (TPU) beschleunigte Instanzen. Bei TPUs handelt es sich um Hardware, die speziell für das Training Neuronaler Netze optimiert wurde. Die in Neuronalen Netzen häufig benötigte Rechenoperation der Matrixmultiplikation kann mit dieser Hardware sehr effizient ausgeführt werden [39]. Die absolute Trainingszeit kann durch Einsatz sowohl von GPUs als auch TPUs stark verkürzt werden. Für das Training der Modelle zur Chipdetektion wurde die mit einer Tesla K80 GPU beschleunigte *Google Colab* Umgebung gewählt. Der GPU stehen in dieser Konfiguration 12 GB RAM zur Verfügung.

Aufgrund des beschränkten GPU-Speichers und der Größe des Modells wurde das *Faster R-CNN* mit einer *Batch Size* von 1 trainiert. Dadurch wird bereits nach jedem Durchlauf eines Trainingsbildes direkt ein Optimierungs- bzw. Gradientenabstiegsschritt durchgeführt. Dies kann in einem, je nach Varianz der Trainingsbilder, unruhigen Trainingsverlauf resultieren. Da das *SSD* Modell deutlich kleiner ist, war ein Training mit einer *Batch Size* von 8 möglich. Dabei wird der Gradient über eine Gruppe von Trainingsbildern, in diesem Fall 8 Stück, gemittelt und mit dem so berechneten, durchschnittlichen Gradienten ein Optimierungsschritt durchgeführt. Dadurch ist mit einem deutlich ruhigeren Trainingsverlauf zu rechnen.

Alle Modelle wurden solange trainiert, bis keine sichtbaren Veränderungen in der mAP auf den zurückgehaltenen Evaluationsbildern zu verzeichnen waren. An diesen Stellen wies auch der kumulierte Fehler des Modells keine wesentlichen Veränderungen mehr auf. Die Entwicklung

⁴<https://colab.research.google.com>

⁵<https://cloud.google.com/compute>

⁶<https://jupyter.org>

Größenkategorie	min. Objektgröße in Pixel ²	max. Objektgröße in Pixel ²
<i>small</i>	0	< 32 ²
<i>medium</i>	32 ²	< 96 ²
<i>large</i>	96 ²	< 10000 ²

Tabelle 2.1.: Kategorien für Objektgrößen

des mAP-Wertes während dem Modelltraining zeigen die Kurven in Anhang A. Der mAP-Wert ist zusätzlich für unterschiedliche Objektgrößen genauer aufgeschlüsselt. Objekte in den Evaluationsbildern werden dabei entsprechend Tabelle 2.1 nach Größe der Bounding Boxen in drei Kategorien eingeteilt.

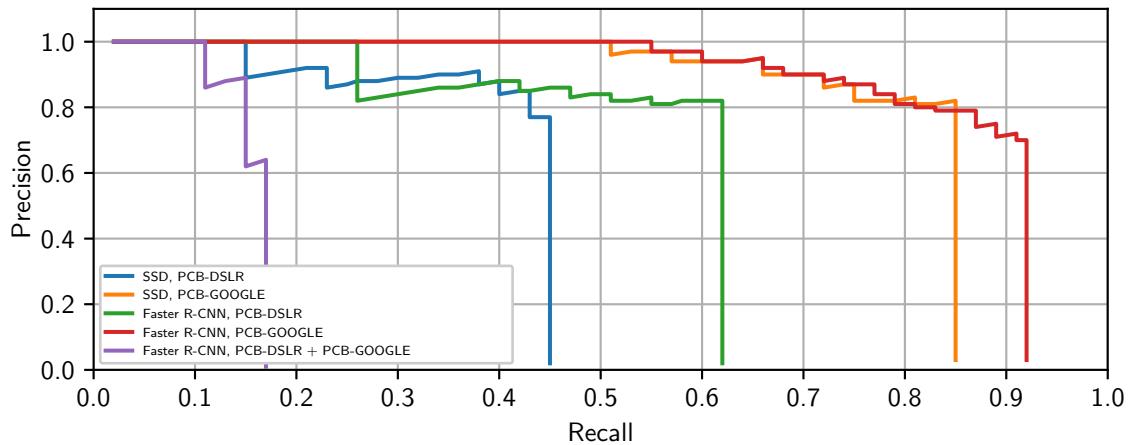
2.3.4. Evaluation

In diesem Kapitel wird zunächst die verwendete Evaluationsmethodik beschrieben. Anschließend werden die von den untersuchten und trainierten Modellen erzielten Ergebnisse vorgestellt. Dabei werden die verschiedenen Neuronalen Netzen untereinander verglichen. Weiterhin wird der Einfluss der verwendeten Trainingsdatensätze auf die Ergebnisse diskutiert. Abschließend wird ein Vergleich zu anderen, verwandten Arbeiten gezogen.

Methodik

Zur Evaluation der untersuchten Verfahren wurde die in Kapitel 2.2.2 vorgestellte Metrik der *Pascal VOC Challenge* genutzt. Teilnehmern des Wettbewerbs wird eine Implementierung dieser Metrik in Form eines Matlab-Skripts bereitgestellt. Es existieren jedoch auch andere Projekte, die diese Metrik implementieren. Eines dieser Projekte ist das auf GitHub veröffentlichte Projekt *Object-Detection-Metrics* [40]. Es handelt sich dabei um eine Reimplementierung der *Pascal VOC*-Metrik in der Sprache Python. Das Projekt verfügt über eine sehr einfache Schnittstelle, basierend auf Textdateien mit den *Ground Truth*-Daten und Detektionsergebnissen, zur Berechnung der Metrik. Damit ist eine Evaluation unabhängig von der gewählten Programmiersprache möglich. Aufgrund dieser Tatsache wurde dieses Projekt für die Evaluation der im Rahmen dieser Arbeit untersuchten Objektdetektoren verwendet. Die für die finale Bewertung herangezogenen Ergebnisse wurden dabei auf dem *PCB-CUSTOM*-Datensatz erzeugt.

Neben den Kennwerten der *Pascal VOC*-Metrik wurde außerdem die über den verwendeten Evaluationsdatensatz gemittelte Detektionszeit bestimmt. Die Messung der Ausführungszeiten erfolgte dabei auf einem Rechner mit einer Intel(R) Core(TM) i5-3320M CPU bei einer Taktrate von 2,60 GHz und 8 GB RAM ohne GPU Beschleunigung.

Abbildung 2.6.: *Precision-Recall*-Kurven der trainierten Modelle

Ergebnisse

Die erzielten Ergebnisse für die AP der Klasse *chip* sind zusammen mit den über die Evaluation gemittelten Ausführungszeiten der Neuronalen Netze in Tabelle 2.2 zusammengetragen.

Model	Trainingsdatensatz	AP für die Klasse <i>chip</i> in %	Detektionszeit in Sekunden
SSD	PCB-DSLR	42,11	2,8
SSD	PCB-GOOGLE	81,52	2,8
Faster R-CNN	PCB-DSLR	57,11	30,1
Faster R-CNN	PCB-GOOGLE	87,38	30,0
Faster R-CNN	PCB-DSLR + PCB-GOOGLE	15,89	29,7

Tabelle 2.2.: Evaluationsergebnisse der trainierten Neuronalen Netze

Die auf dem *PCB-GOOGLE* Datensatz trainierten Modelle zeigen mit 81,52 % für das SSD bzw. sogar 87,38 % AP für das *Faster R-CNN* Modell eine insgesamt sehr gute Objekterkennungsleistung. Dies bestätigt auch die in Abbildung 2.6 gezeigten *Precision-Recall*-Kurven. Offensichtlich ist der *PCB-DSLR* Datensatz weniger gut für das Training Neuronaler Netze zur Erkennung einzelner Chips geeignet, was aber auch mit den gewählten Evaluationsbildern zusammenhängt. *PCB-CUSTOM* enthält überwiegend Nahaufnahmen von recht kleinen PCBs mit wenigen Chips und entspricht daher eher Aufnahmen des *PCB-GOOGLE* Datensatzes. Bilder des *PCB-DSLR* Datensatzes zeigen hingegen größere Platinen mit vielen verbauten Komponenten.

Wie in den *Precision-Recall*-Kurven aus Abbildung 2.6 zu erkennen ist, liefern beide untersuchten Modelle sehr hochwertige Detektionen. Die *Precision*-Werte befinden sich durchweg auf einem sehr hohen Niveau bis zum Punkt des jeweils maximalen *Recall*-Wertes. Das *Faster*

R-CNN Modell konnte unabhängig vom gewählten Trainingsdatensatz im direkten Vergleich mit dem *SSD* Modell eine um +5,86 % bzw. auf dem *PCB-DSLR* sogar eine um +15 % bessere AP erreichen. Bei dem *PCB-DSLR* Datensatz profitiert das *Faster R-CNN* Modell von der Verarbeitung höher aufgelöster Bilder in den niedrigeren Schichten, wodurch es in der Lage ist, auch kleinere Objekte noch zu erkennen. In Aufnahmen, die auf die fixen Dimensionen des Inputs für das *SSD*-Modells herunterskaliert wurden, ist es auch für Menschen schwierig, die teils sehr kleinen Chips noch sicher zu detektieren. Aufgrund des komplexeren Aufbaus benötigt das *Faster R-CNN* Modell eine um den Faktor 10,8 längere Berechnungszeit als das *SSD* Modell.

Vergleich mit verwandten Arbeiten

Kleber et al. konnten durch ihr vorgestelltes Verfahren zur Chiplokalisierung mittels Farbsegmentierung und anschließender Anwendung des *Watershed*-Algorithmus auf ihrem Bilddatensatz eine Erkennungsrate von über 65 % erzielen [9]. Leider stand der in dieser Arbeit verwendete Datensatz zur Evaluierung der trainierten Neuronalen Netze nicht zur Verfügung. Dennoch ist zu vermuten, dass Neuronale Netze den auf Farbsegmentierung basierenden Verfahren überlegen sind, wie Abbildung 2.7 deutlich macht: Der Lötstopplack des dargestellten PCBs besitzt nahezu die gleiche Farbe wie die *Packages* der Chips. Folglich ist es schwer, nur aufgrund der Farbinformation zwischen einem Chip und dem PCB selbst zu unterscheiden. Beide untersuchten Neuronalen Netze sind aber in der Lage, zuverlässig die einzelnen Chips zu erkennen, da sie Merkmale, wie beispielsweise Farbübergänge oder auch Farben selbst, in unterschiedlichen Hierarchiestufen für die Objekterkennung nutzen können. In dem in Abbildung 2.7 gezeigten PCB wird lediglich ein einzelner Chip durch das *Faster R-CNN* und zwei Chips von dem *SSD* Modell nicht erkannt. FP treten in beiden Modellen nicht auf.

2.4. Pin-Erkennung

Der zweite zur automatisierten Sicherheitsanalyse einer Platine essentielle Schritt bei der visuellen Analyse ist die zuverlässige und exakte Lokalisierung einzelner Komponentenpins. Ziel ist es, hierbei möglichst präzise die Mittelpunktkoordinaten der Stellen zu bestimmen, an denen die Komponentenpins auf das PCB aufgelötet wurden. Pins sind in diesem Bereich flach und verlaufen parallel zur PCB-Oberfläche. Die Wahrscheinlichkeit, dass eine gesetzte Prüfnadel ungewollt abrutscht, ist an diesen Stellen, im Gegensatz zu den gebogenen Pins selbst, geringer. Sie eignen sich daher besser für eine elektrische Kontaktierung.

Als Ausgangspunkt für die Pinerkennung dienen die zuvor mittels des in Kapitel 2.3 vorgestellten Verfahrens bestimmten Bounding Boxen der einzelnen Komponenten. Weiterhin wird vorausgesetzt, dass eventuell vorhandene Rotationen der Chips innerhalb der Bounding Boxen entfernt wurden. Die Kanten der Chips verlaufen damit parallel zu den Kanten der Boxen. Basierend auf diesen Annahmen wurde ein Verfahren zur automatischen Lokalisierung von

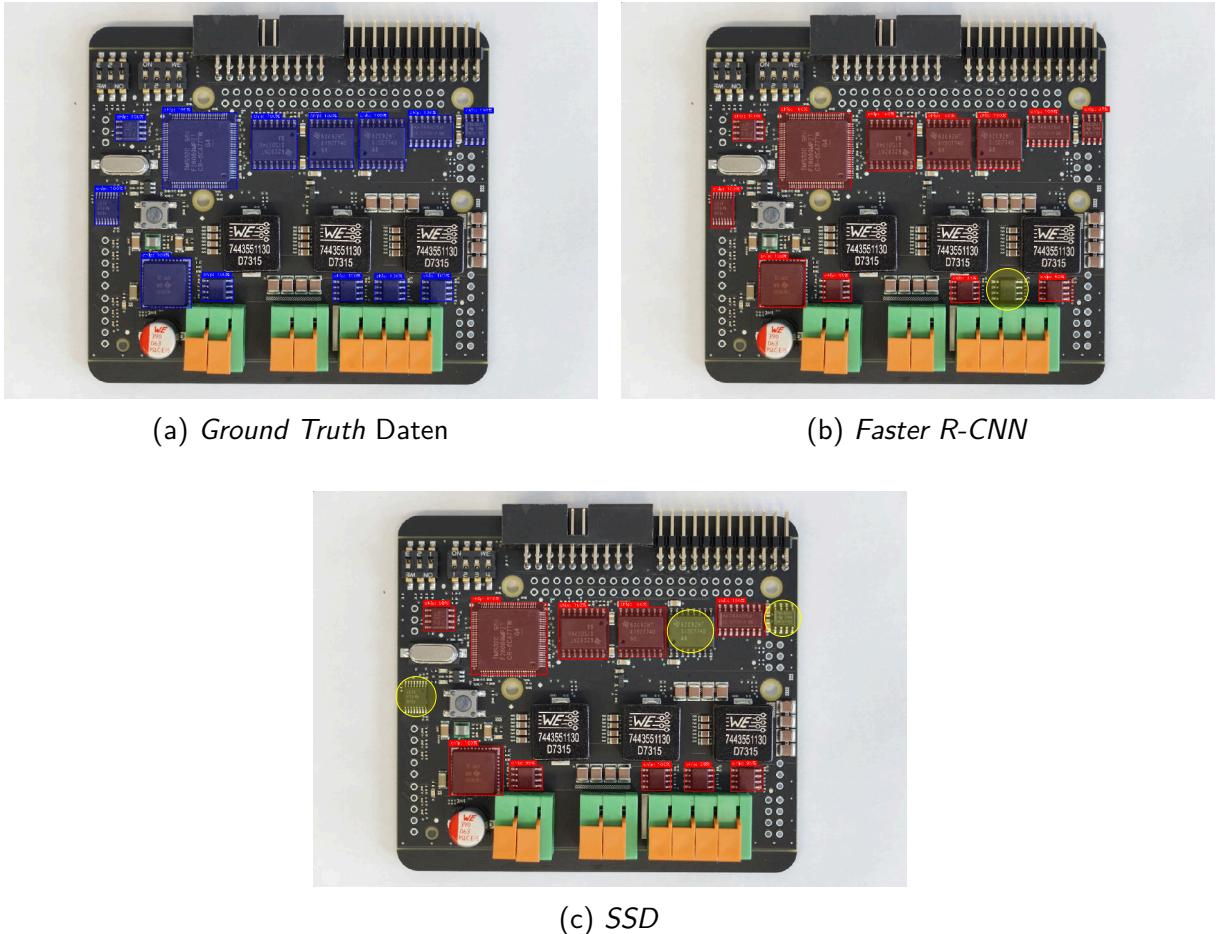


Abbildung 2.7.: Vergleich der *Ground Truth* Daten und den erkannten Chips der auf dem *PCB-GOOGLE* Datensatz trainierten Neuronalen Netze bei einem schwarzen PCB (*PCB-CUSTOM*, Aufnahme DSC00627); Blau: *Ground Truth*-Daten, Rot: korrekt erkannte Chips, Gelb: nicht erkannte Chips

Komponentenpins entwickelt, das in diesem Kapitel vorgestellt wird. Wie bereits zur Evaluation und Training der Neuronalen Netze für die Chiperkennung wurde auch für die Pinerkennung ein Bilddatensatz erstellt. Dieser Datensatz ist Gegenstand von Kapitel 2.4.1. In Kapitel 2.4.2 wird anschließend das in Form einer Bilderverarbeitungspipeline entwickelte Verfahren zur Pinlokalisierung beschrieben. Zum Vergleich wurden außerdem das bereits aus Kapitel 2.3 bekannte *Faster R-CNN* Modell und das *SSD* Modell auf dem Datensatz für die Pinerkennung trainiert. In Kapitel 2.4.3 werden die genutzten Modellkonfigurationen und gewählten Trainingsparameter vorgestellt. Die zur Bewertung der Pindetektionsleistung und dem Vergleich der untersuchten Verfahren eingesetzten Kennzahlen sind Gegenstand von Kapitel 2.4.4. Eine Zusammenfassung der erzielten Ergebnisse und eine abschließende Diskussion findet sich in Kapitel 2.4.4.

2.4.1. Bilddatensatz

Für die Einstellung der Parameter der entwickelten Bildverarbeitungspipeline und zur späteren Evaluation wurde ein Bilddatensatz, der im weiteren als *CHIP-EVAL* bezeichnet wird, aufgebaut. Er besteht aus Bildaufnahmen, die jeweils nur einen einzigen, auf einem PCB verbauten Chip zeigen. Zur Erstellung des Datensatzes wurden die im *PCB-CUSTOM* gelabelten Chips entsprechend ihrer Bounding Boxen mit Zugabe eines 10 Pixel breiten Randes ausgeschnitten. Der somit erzeugte Datensatz besteht aus insgesamt 48 Bildern einzelner Chips. Es sind überwiegend SMD-Chips im Thin Quad Flat Package (TQFP)-*Package* enthalten, jedoch auch fünf Aufnahmen von Chips mit BGA-*Package*. Bei letzteren befinden sich die Pins auf der Unterseite der Chips. Sie sind daher in den Bildaufnahmen nicht sichtbar. Ein gutes Verfahren zur Pinerkennung sollte daher in diesen Aufnahmen auch keine Ergebnisse liefern.

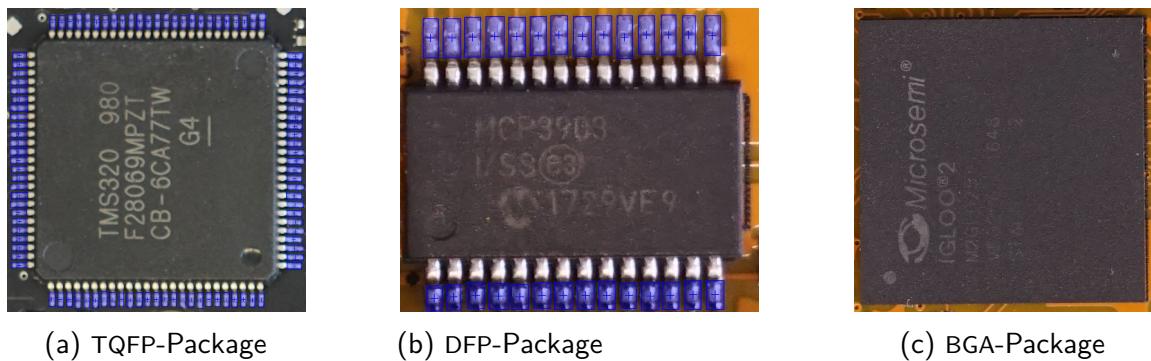


Abbildung 2.8.: Beispielbilder mit blau dargestellten *Ground Truth*-Daten des *CHIP-EVAL*-Datensatzes

Nach der Extraktion der Chipbilder wurden die einzelnen Pins manuell gelabelt. Jeder Pin wurde dabei durch eine rechteckige Bounding Box beschrieben. Die Boxen sind so gewählt, dass sie den Bereich, an dem der Pin an das PCB angelötet wurde, einschließen. Der Punkt zur optimalen Kontaktierung befindet sich damit in der Mitte der Bounding Boxen. In Abbildung 2.8 sind einige Beispielbilder des Datensatzes dargestellt. Die gelabelten Pins sind darin als blau hinterlegte Rechtecke eingezeichnet. Die Zentren der Bounding Boxen sind mit einem Kreuz markiert. In dem Datensatz sind insgesamt 1181 gelabelte Pins enthalten. Die Anzahl der Pins pro Chip variiert dabei je nach Chip *Package* von 6 bis 100. BGA-Chips sind hierbei ausgenommen.

2.4.2. Computer Vision Pipeline zur Pinerkennung

ICs weisen eine sehr regelmäßige Struktur auf. Die auf einem *Die* aufgebrachten, integrierten Schaltkreise sind meist mit einem Gehäuse aus Kunststoff oder Metall zur Ableitung der beim Betrieb entstehenden Abwärme bedeckt. Der eingesetzte Kunststoff ist hierbei meistens schwarz. Auf dem Gehäuse ist oftmals eine herstellerspezifische Identifikationsnummer aufgedruckt, die den Chip eindeutig kennzeichnet. Die Positionen der Pins, mit denen der Chip auf ein PCB

aufgelötet wird, variiert je nach Chip *Package*. Bei SMD-Bauteilen befinden sich die meist silberfarbenen Pins an den Seiten des Chipgehäuses. Auch die Anzahl der Pins und Seiten, an denen diese angebracht sind, sind je nach Bauform unterschiedlich. Bei DFP finden sich Pins nur an den beiden Längsseiten der Chips. Triple Flat Pack (TFP)-Chips besitzen Pins an drei Seiten und Quad Flat Pack (QFP) sogar an allen vier Gehäuseseiten. Der Platz auf PCBs ist meist sehr beschränkt. Im Zuge der immer fortschreitenden Miniaturisierung wurde daher das BGA-Package entwickelt. Bei diesem liegen die Pins nicht als Metallfüßchen an den Chipgehäusen vor, sondern sind auf die Unterseite des Chips als Lötpolen verlagert. Die mittlerweile kaum noch eingesetzten THT-Packages, bei denen die Bauteilpins durch das PCB gesteckt und verlötet werden, wurden von den weiteren Betrachtungen in dieser Arbeit ausgeschlossen [5, S. 15].

Lediglich Pins von Chips mit SMD-*Package* können ohne eine Demontage der Bauteile bei einer Sicherheitsanalyse eines eingebetteten Systems direkt kontaktiert werden. Das nachfolgend vorgestellte Verfahren beschränkt sich daher auf die Erkennung von Pins dieser Gehäuseform. Aufgrund der klaren geometrischen Anordnung und Farbgebung der Pins wurde eine klassische Bildverarbeitungspipeline zur Lokalisierung von diesen in vorsegmentierten Bildaufnahmen entwickelt. Es wird im Weiteren als *COLOR-SEG*-Verfahren bezeichnet. Die dabei durchlaufenen Schritte sind schematisch in Abbildung 2.9 dargestellt. Nachfolgend wird auf die einzelnen Schritte im Detail eingegangen. Die Ergebnisse der jeweiligen Zwischenschritte sind in Abbildung 2.10 anhand einer Bildaufnahme aus dem *CHIP-EVAL*-Datensatz visualisiert.

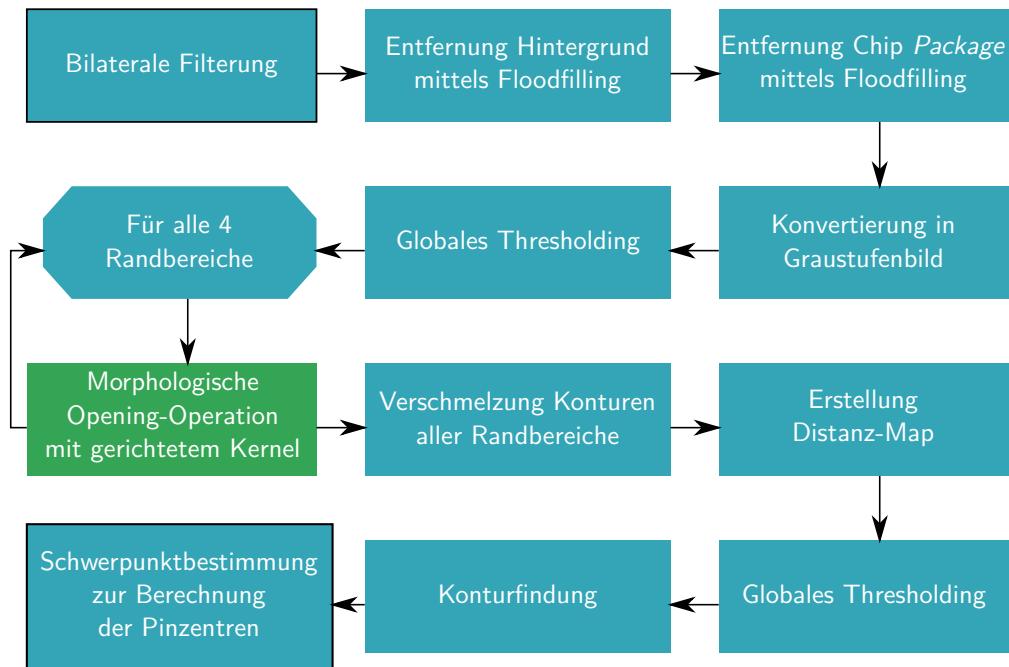


Abbildung 2.9.: Computer Vision Pipeline zur automatischen Pin-Erkennung

Zunächst wird das Eingangsbild mittels der in Kapitel 2.2.6 vorgestellten bilateralen Filterung geglättet und von eventuell vorhandenem Bildrauschen befreit. Die vorsegmentierten Chipaufnahmen können noch größere Bereiche des PCBs enthalten, die sich je nach Farbe des Löstopplacks negativ auf die Pinerkennung auswirken können. Daher wird im nächsten Schritt

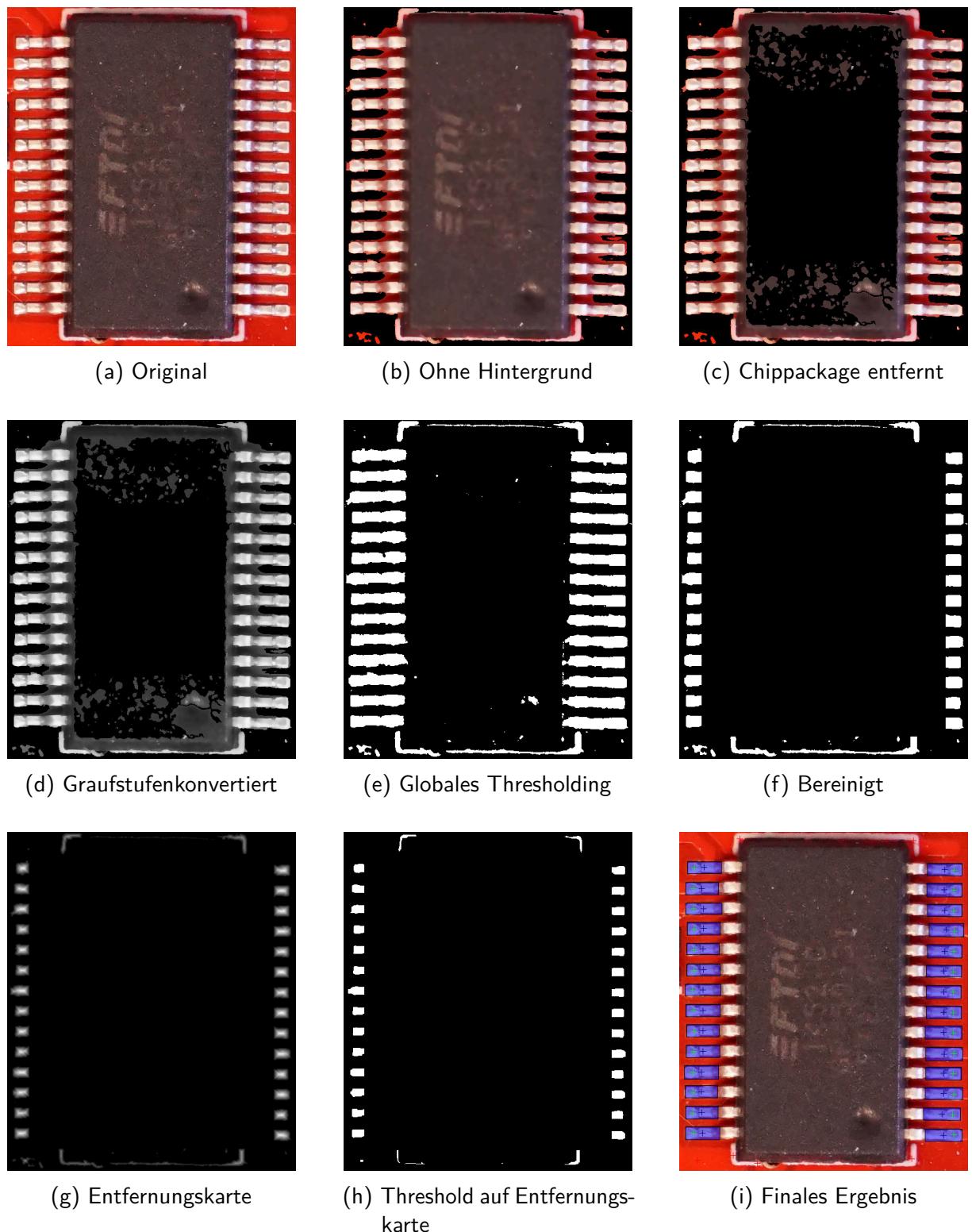


Abbildung 2.10.: Zwischenschritte der Computer Vision Pipeline zur Pinerkennung

mittels des *Floodfill*-Verfahrens eine Entfernung des verbliebenen PCB-Hintergrunds durchgeführt (siehe Kapitel 2.2.5). Als *Seed*-Punkte werden dabei die Randpixel der Bildaufnahme genutzt, da diese mit großer Wahrscheinlichkeit die Farbe des Lötstopplacks besitzen. Das *Floodfill* wird im RGB-Farbraum mit einem fixen Farbbereich von ± 40 Einheiten in jedem Farikanal bezogen auf die Farbe des ursprünglichen *Seed*-Pixels durchgeführt. Das Ergebnis dieser Hintergrundentfernung zeigt Abbildung 2.10b.

Anschließend wird der Bereich des Chip *Packages* entfernt. Dieser Schritt wurde eingeführt, da die Bereiche der aufgedruckten IDs auf dem Chipgehäuse eine sehr ähnliche Farbe wie die zu detektierenden Pins aufweisen und oftmals fälschlicherweise als Pins erkannt wurden. Zur Entfernung des Chip *Packages* wird ebenfalls der *Floodfill*-Algorithmus angewendet. Es wird angenommen, dass sich der Chip in der Mitte des Bildes befindet. Um sowohl das Chipgehäuse als auch die Beschriftung der Chip-ID durch das *Floodfill* zu erfassen, werden als *Seed* die Punkte in einem rechteckigen, um die Mitte des Bildes zentrierten Bereichs gewählt. Die Höhe und Breite dieses Bereiches werden prozentual bezogen auf die Bildabmessungen bestimmt. Mit einem sowohl für die Höhe als auch Breite identischen Wert von 50 % konnten hierbei gute Ergebnisse erzielt werden (vgl. Abbildung 2.10c).

Das verbliebene Bild enthält nun primär die Komponentenpins sowie nicht vollständig entfernte Bereiche des Chip *Packages* und des PCBs. Es wird angenommen, dass die Pins aufgrund ihrer silbernen Farbe die hellsten Bereiche des Bildes darstellen. Unter dieser Annahme wird eine Graustufenkonvertierung und anschließend ein globales Thresholding durchgeführt, um die Pinflächen zu extrahieren (siehe Abbildungen 2.10d und 2.10e). Als Ergebnis liegt eine binäre Maske vor, die überwiegend Pinregionen enthält.

Das binäre Bild enthält nach der Schwellwertbildung meist noch kleine Konturen. Je nach Aufnahmewinkel und Qualität der Lötstellen kann es außerdem auftreten, dass aufgrund zu geringer Farbunterschiede einzelne Pins über dünne Stege in der binären Maske miteinander verbunden sind. Zur Bereinigung des Bildes und besseren Segmentierung der einzelnen Pins werden die in Kapitel 2.2.4 beschriebenen, morphologischen Operationen eingesetzt. Pins sind an den Chipgehäusen in einem regelmäßigen Raster angebracht. Um diesem Sachverhalt Rechnung zu tragen, werden die morphologischen Operationen jeweils in lokalen Bereichen mit einem gerichteten, rechteckigen Kernel eingesetzt. Dabei werden vier Randzonen des Bildes separat betrachtet. Als Randzone ist dabei ein Bereich mit fixem Abstand zum Bildrand entlang einer Bildkante zu verstehen. Als Abstand wird dabei ein fixer Wert angenommen, der aktuell der Breite des zusätzlichen Randes bei der Extraktion des Chipbildes aus der PCB-Aufnahme und einem zusätzlichen Offset von 30 Pixeln entspricht. Im oberen und unteren Randbereich wird für die morphologischen Operationen ein senkrecht stehender Kernel mit den Abmessungen 3×1 und ein entsprechend um 90° gedrehter 1×3 Kernel für die linke und rechte Randzone verwendet. Mit diesen Kernen werden jeweils drei morphologische *Closing*-Operationen zur Bereinigung des Bildes durchgeführt. Für die weiteren Schritte des Verfahrens werden nur die verbliebenen Konturen in den jeweiligen Randbereichen genutzt. Das Ergebnis dieses

Verarbeitungsschrittes ist in Abbildung 2.10f zu sehen. Wie zu erkennen ist, wird durch die Beschränkung auf Randzonen fixer Breite ein Teil der Pinkonturen abgetrennt. Lediglich die äußeren Enden verbleiben in der finalen Maske. Bei diesen handelt es sich überwiegend um die Bereiche, an denen die Pins auf dem PCB aufgelötet wurden. Sie entsprechen damit genau den für die spätere elektrische Kontaktierung relevanten Pinbereichen.

Eine finale Filterung findet durch Berechnung einer Distanz-Map und anschließendem globalen Thresholding auf dieser statt. Bei der in Kapitel 2.2.7 beschriebenen *Distance Transform* zur Erzeugung einer Distanz-Map wird die kürzeste Distanz zum nächsten Hintergrundpixel bestimmt. Zentren großflächiger Vordergrundareale erhalten hierbei größere Werte als zum Beispiel linienförmige Konturen (vgl. Abbildung 2.10g). Durch Anwendung eines globalen Thresholdings auf die normalisierte Distanz-Map verbleiben in der finalen binären Maske nur Zentren großflächiger, vorzugsweise quadratischer Areale, wie Abbildung 2.10h zeigt.

Abschließend wird eine Suche von Zusammenhangskomponenten durchgeführt (siehe Kapitel 2.2.8). Die Konturen der damit bestimmten Bereiche bilden die detektierten Komponentenpins. Als Pinzentrum wird der Schwerpunkt dieser Konturen gesetzt. Das Ergebnis ist in Abbildung 2.10i zu sehen. Die blauen Bereiche markieren darin die *Ground Truth*-Boxen der gelabelten Pins. Grün eingefärbte Kreuze zeigen korrekt erkannte TP Pins. FP sind in rot dargestellt.

2.4.3. Pinerkennung mit Neuronalen Netzen

Mit den in Kapitel 2.3 vorgestellten technischen Neuronalen Netzen konnten bereits sehr gute Ergebnisse bei der Chiperkennung erzielt werden. Pins eines Chips können ebenfalls als Objekte in Form einer Bounding Box beschrieben werden (vgl. Kapitel 2.4.1). Es liegt daher nahe, die bereits genutzten Modelle auch auf dem in Kapitel 2.4.1 vorgestellten Bilddatensatz zur Pinerkennung zu trainieren.

Dazu wurden dieselben Pipelines und Konfigurationen der *Tensorflow Object Detection API* genutzt, wie sie bereits in Kapitel 2.3.3 beschrieben wurden. Für das Training wurde der *CHIP-EVAL* Datensatz in einem Verhältnis von 1:4 in Bilder für die Evaluation und das Training aufgeteilt. Besonders schwierige, weil sehr verschwommen oder unter schlechten Lichtverhältnissen aufgenommene Bilder, wurden nicht für das Training der Neuronalen Netze verwendet. Somit standen insgesamt 38 Aufnahmen für das Training und neun Bildaufnahmen zur Evaluation während des Trainingsprozesses zur Verfügung. Das Training erfolgte ebenfalls auf der *Google Colaboratory*-Plattform. Die Entwicklung der mAP während des Trainings beider Modelle findet sich in Anhang B.

2.4.4. Evaluation

Die Pinerkennung kann als vereinfachte Aufgabe einer Objekterkennung betrachtet werden. Kennwerte der bereits bei der Chiperkennung eingesetzten Evaluationsmethodik können daher

auch hier in abgewandelter Form eingesetzt werden. Eine genaue Beschreibung der verwendeten Kennwerte zur Bewertung der entwickelten Verfahren gibt der nächste Abschnitt. Anschließend werden die erzielten Ergebnisse der einzelnen untersuchten Pindetektoren vorgestellt und ausführlich diskutiert.

Methodik

Zur Beurteilung der untersuchten Verfahren zur Pinerkennung wurden vier Kennwerte herangezogen. Die bereits in Kapitel 2.2.2 vorgestellten Kennzahlen *Precision* und *Recall* werden auch im Kontext der Pinerkennung wiederverwendet. Dabei wird ein detekterter Pin als TP eingestuft, falls sich der detektierte Pinmittelpunkt innerhalb einer der *Ground Truth*-Bounding Boxen befindet. Liegen mehrere detektierte Pins in einer *Ground Truth*-Box, so wird der erste Pin in der Ergebnisliste als TP, alle weiteren als FP kategorisiert. Mit dieser Einteilung lassen sich die Kennzahlen *Precision* und *Recall* auch für die Pinerkennung anwenden. Die über den gesamten Evaluationsdatensatz gemittelten Werte werden im Weiteren als *mean Precision* (mP) und *mean Recall* (mR) bezeichnet um Verwechslungen mit dem Kennwert der mAP der *Pascal VOC*-Metrik zu vermeiden.

Neben der reinen Erkennung eines Pins sind für die anschließende, automatisch durchgeführte, elektrische Kontaktierung insbesondere die exakten Koordinaten der Pinzentren von Bedeutung. Um diese Eigenschaft eines Pindetektors beurteilen zu können, wurde ein Maß für den Lokalisierungsfehler eingeführt. Der Kennwert ist dazu wie folgt definiert: Sei eine Bildaufnahme mit einer Menge gelabelter Pins $P_{gt} \subseteq \mathbb{R} \times \mathbb{R} \times \mathbb{R}^+ \times \mathbb{R}^+$ gegeben. Die Fläche eines einzelnen Pins $p_{gt} \in P_{gt}$ ist durch die Zentrumskoordinaten $x_{gt}, y_{gt} \in \mathbb{R}$, die Breite $h_{gt} \in \mathbb{R}^+$ und Höhe $w_{gt} \in \mathbb{R}^+$ in Bildkoordinaten definiert. Weiterhin seien P_{Det} die Menge der als TP klassifizierten Detektionen eines Pindetektors für diese Bildaufnahme. Ein detekterter Pin $p_{Det} \in P_{Det}$ wird dabei durch die Zentrumskoordinaten $x_{Det}, y_{Det} \in \mathbb{R}$ beschrieben. Dann berechnen sich die relativen Abstände zum Rand der Pinfläche d_x und $d_y \in \mathbb{R}$ in Bildkoordinaten zu

$$d_x(p_{Det}, p_{gt}) = 1 - \left| \frac{x_{Det} - x_{gt}}{0.5 * w_{gt}} \right|$$

$$d_y(p_{Det}, p_{gt}) = 1 - \left| \frac{y_{Det} - y_{gt}}{0.5 * h_{gt}} \right|$$

Bereits das Überschreiten der Pinfläche in einer Bilddimension kann für nachfolgende Schritte, wie die elektrische Kontaktierung, gravierende Folgen haben. Um diesem Sachverhalt Rechnung zu tragen, werden die relativen Randabstände anschließend multiplikativ zu dem neu definierten Kennwert der *Relative Border Distance* (RBD) verknüpft.

$$RBD(p_{Det}, p_{gt}) = d_x(p_{Det}, p_{gt}) \cdot d_y(p_{Det}, p_{gt})$$

Wird der Mittelpunkt eines Pins exakt getroffen, wird ein RBD von 1 erzielt. Liegen die erkannten

Pinkoordinaten auf dem Rand der Pinfläche, liegt ein RBD von 0 vor. Durch die multiplikative Verknüpfung genügt es, wenn diese Bedingung für eine einzige Bilddimension zutrifft um einen RBD von 0 zu erzeugen. Auch für diesen Kennwert wurde ein über einen Bilddatensatz I mit $n \in \mathbb{N}^+$ gelabelten Bildern gemittelter Wert mit den TP Detektionen $P_{\text{Det},i}$ für ein Bild $i \in I$ bestimmt. Dieser wird nachfolgend als *mean Relative Border Distance* (mRBD) bezeichnet und wie folgt berechnet

$$\text{mRBD} = \frac{1}{n} \sum_{i \in I} \sum_{p_{\text{Det},i} \in P_{\text{Det},i}} \text{RBD}(p_{\text{Det},i}, p_{gt,i})$$

Zur Beurteilung, wie rechenintensiv ein Verfahren für die Pindetektion ist, wurde als letzte Kennzahl die mittlere Detektionsdauer $\overline{t_D} \in \mathbb{R}^+$ berechnet. Hierbei handelt es sich um den Mittelwert der Detektionszeiten $t_i \in \mathbb{R}^+$ für ein Bild i aus dem Evaluationsdatensatz.

$$\overline{t_D} = \frac{1}{n} \sum_{i=0}^n t_i$$

Ergebnisse

Eine Übersicht der erzielten Kennwerte aller untersuchter Verfahren gibt Tabelle 2.3. Die Ergebnisse wurden jeweils mit den auf den *CHIP-EVAL* Datensatz optimierten Verfahren erzielt. Die beiden Neuronalen Netzen wurden, wie in Kapitel 2.4.3 beschrieben, bereits auf dem Datensatz trainiert. Bis auf die neun zurückgehaltenen Bilder konnten die Modellparameter entsprechend auf diesen Datensatz optimiert werden. Bei dem in Kapitel 2.4.2 beschriebenen *COLOR-SEG* Verfahren wurden die wichtigsten Parameter händisch auf den Datensatz abgestimmt.

Verfahren	mP in %	mR in %	$mRBD$	$\overline{t_D}$ in Sekunden
COLOR-SEG	72,5	80,6	0,441	0,350
Faster R-CNN	86,3	77,9	0,614	29,314
SSD	55,5	21,4	0,163	2,695

Tabelle 2.3.: Erzielte Ergebnisse bei der Pinerkennung

Mit einer mP von 55,5 % und lediglich 21,4 % für den durchschnittlichen *Recall* zeigt das *SSD* Modell das schlechteste Ergebnis. Bereits Liu et al. weisen in ihrer Arbeit auf die Schwäche des Modells bei der Erkennung von kleinen Objekten hin [14]. Dies ist insbesondere bei Chips mit QFP-Package der Fall. In Aufnahmen von Chips diesen Typs sind bis zu 100, bezogen auf die Gesamtbildgröße, sehr kleine Pins enthalten. Auch die wenigen, korrekt erkannten Pins haben nur eine geringe Übereinstimmung mit den *Ground Truth*-Daten, was sich durch einen mit 0,163 recht kleinen mRBD bemerkbar macht.

Im Gegensatz dazu konnten mit dem *Faster R-CNN* Modell, wie bereits bei der Chiperkennung, sehr gute Ergebnisse bei der Erkennung einzelner Pins erreicht werden. Es erzielt mit

einem durchschnittlichen *Recall*-Wert von 77,9 % eine sehr hohe Detektionsrate auf dem Evaluationsdatensatz. Die Detektionen sind mit einem mRBD von 0,614 außerdem sehr hochwertig. Wie der mP von 86,3 % zu entnehmen ist, werden durch das Verfahren nur relativ wenige FP-Ergebnisse geliefert. Die Ausführungszeit von 29,314 s ist jedoch vergleichsweise lange.

Auch die entwickelte Computer Vision Pipeline liefert mit einer mP von 72,5 % und einem mRBD von 0,441 gute Pindetektionen. Mit einem mR von 80,6 % konnte mit diesem Verfahren sogar die höchste durchschnittliche Erkennungsrate erzielt werden. Es ist mit einer durchschnittlichen Ausführungszeit von 0,35 s zugleich das schnellste aller untersuchten Verfahren.

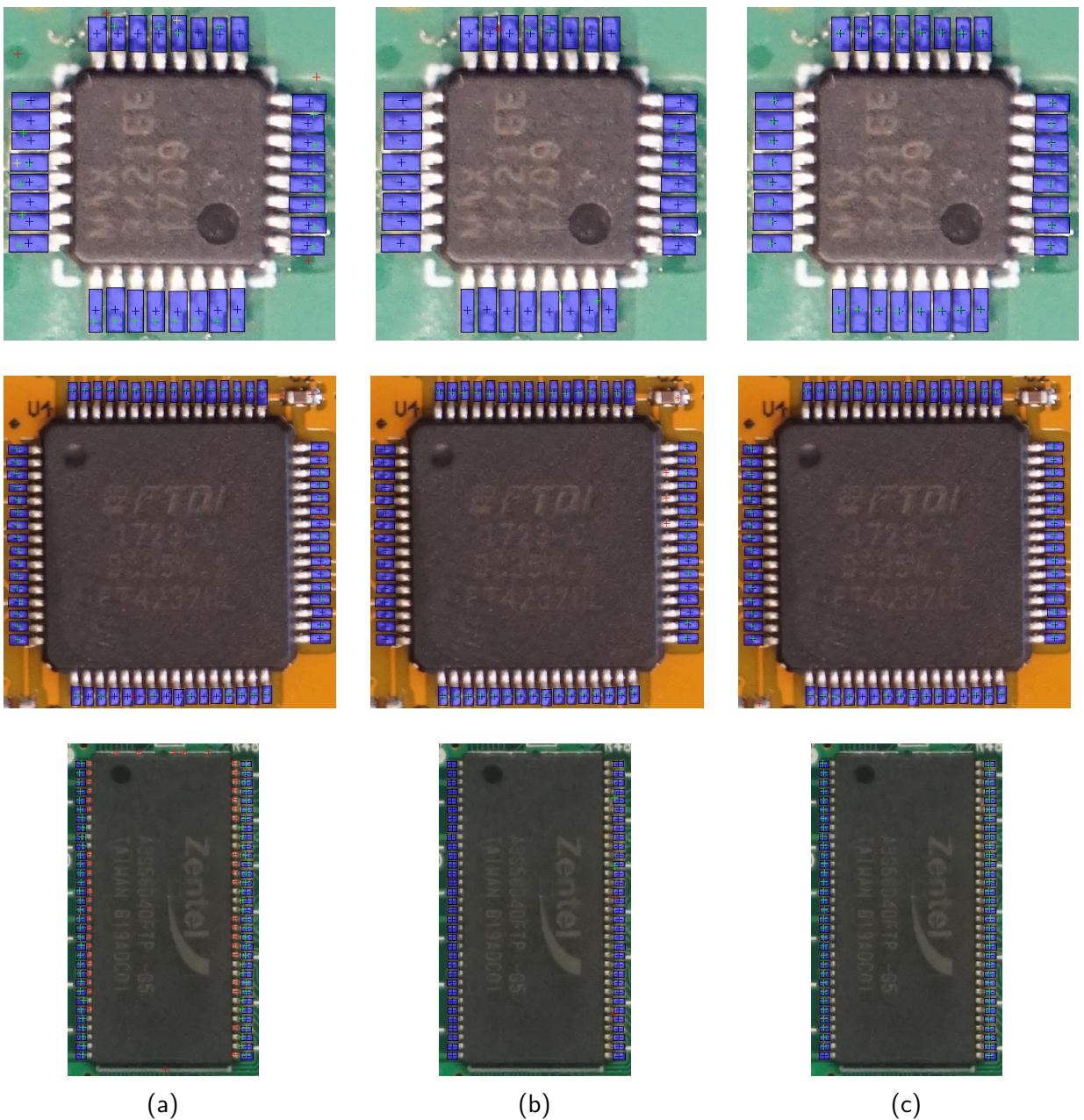


Abbildung 2.11.: Detektionsergebnisse der Verfahren *COLOR-SEG* a), *SSD* b), *Faster R-CNN* c) bei drei unterschiedlichen Chips; Grün: Korrekt erkannter Pin, Rot: Falsch erkannter Pin, Gelb: Mehrfach erkannter Pin, Blau: *Ground Truth*-Daten

Einige Fehlerquellen der Verfahren, insbesondere der aufgebauten Computer Vision Pipeline, werden anhand der in Abbildung 2.11 dargestellten Detektionsergebnisse auf dem *CHIP-EVAL* Datensatz abschließend diskutiert. In Abbildung 2.11 sind die *Ground Truth*-Boxen blau eingefärbt. Die Mittelpunkte dieser Boxen sind in Form von blauen Kreuzen eingezeichnet. Die detektierten Pins werden ebenfalls durch Kreuze visualisiert. Dabei sind TP grün, FP rot und mehrfache Detektionen für die selbe *Ground Truth*-Box gelb eingefärbt. Mehrfachdetektionen werden ebenfalls zur Kategorie der FP gezählt.

Das *COLOR-SEG*-Verfahren neigt zu Fehldetektionen an hellen Bereichen im Umfeld der Chips. Dies ist in Abbildung 2.11a in der ersten und zweiten Bildzeile sehr gut zu erkennen. Grund hierfür ist eine unzureichende Hintergrundentfernung im zweiten Schritt der Pipeline. Die Anwendung von morphologischen Operationen auf die durch das *Floodfill* erzeugten Hintergrundmaske könnte hier zu Verbesserungen führen. Ein weiteres in Abbildung 2.11a unten zu erkennendes Problem stellen Mehrfachdetektionen am Ansatz der Pins an dem Chipgehäuse dar. Diese resultieren aus einer Aufsplittung der Pinregionen im fünften Schritt der Pipeline, dem globalen Thresholding auf dem Graustufenbild ohne Hintergrund und Chip *Package*. Die Ursache sind Schattenwürfe im Knick der gebogenen Pins. Ein Weichzeichnen der Übergänge oder ein Thresholding mit einem niedrigeren Schwellwert könnten diese Probleme beseitigen.

Das *SSD* Modell erkennt insgesamt sehr wenige Pins. Dies zeigt sich auch in den Resultaten der Evaluation die (siehe Abbildung 2.11b). Der Grund hierfür liegt in der Architektur des Modells, die bei der Erkennung von kleinen Objekten Probleme bereitet. Weiterentwicklungen des *SSD* Modells versuchen durch die Fusion von Feature-Maps die Erkennungsrate von kleinen Objekten zu erhöhen [41][42]. Diese Probleme treten bei dem trainierten *Faster R-CNN* Modell nicht auf. Es wurde, wie auch das *SSD* Modell auf dem *CHIP-EVAL*-Datensatz trainiert und erkennt nahezu alle in Abbildung 2.11c dargestellten *Ground Truth*-Pins.

2.5. Zusammenfassung und Ausblick

Die erzielten Ergebnisse und gewonnenen Erkenntnisse bei der visuellen PCB-Analyse sind in diesem Abschnitt nochmals zusammengefasst. Basierend auf den gewonnenen Erkenntnissen werden weiterhin Anregungen für Weiterentwicklungen der beschriebenen Verfahren gegeben und mögliche alternative Ansätze skizziert.

Zusammenfassung Chiperkennung

Mit aktuellen technischen Neuronalen Netzen konnten mit einem mAP von bis zu 87,38 % sehr gute Ergebnisse auf einem dem Stand der Technik entsprechenden Bilddatensatz bei der Chiperkennung erzielt werden. Der untersuchte *Multi-Stage*-Detektor *Faster R-CNN* war dabei dem ebenfalls untersuchten *Singel Shot*-Detektor *SSD* bezüglich der Detektionsqualität überlegen. Weiterhin konnte die Wichtigkeit einer großen Varianz der Trainingsdaten durch den

Vergleich zweier Bilddatensätze aufgezeigt werden. Von den genutzten Datensätzen konnten mit dem neu zusammengestellten *PCB-GOOGLE* die besten Resultate erzielt werden.

Alle beschriebenen Verfahren zur Chiperkennung wurden mit einer einheitlichen API als eigenständiges Modul in der Programmiersprache Python unter Nutzung der Tensorflow-Bibliothek implementiert. Die untersuchten Verfahren können damit auch in weiterführenden Arbeiten im Bereich der visuellen PCB-Analyse oder einem übergeordneten System zur automatisierten IT-Sicherheitsanalyse sehr einfach eingebunden werden.

Weiterentwicklungen der Chiperkennung

Um die Detektionsraten bei der Chiperkennung weiter zu verbessern, könnten neuere Netztopologien, wie etwa verbesserte Varianten des *SSD* Modells, auf dem zusammengestellten *PCB-GOOGLE* Datensatz trainiert werden. Auch eine Erweiterung dieses Datensatzes um Aufnahmen aus der realen Praxis eines Sicherheitsanalysten, wie sie bereits im *PCB-CUSTOM*-Datensatz enthalten sind, könnte zu besser angepassten Objektdetektoren führen. Da sich *Chip Packages*, wie *BGA* und *SMD*, in ihrem Aussehen stark unterscheiden, könnte die Einteilung in unterschiedliche Objektklassen den Lernprozess für die Neuronalen Netze zusätzlich vereinfachen. Dies verkürzt die Trainingszeiten und führt vermutlich insgesamt zu besseren Detektionsergebnissen. Mit weiteren Objektkategorien könnten zusätzlich auch passive Baulemente, wie etwa Widerstände, Spulen oder Kondensatoren, automatisch erkannt werden. Dies könnte zu einem vollständigeren Bild und damit zu einem besseren Verständnis des zu untersuchenden PCBs führen.

Zusammenfassung Pinerkennung

Für die Aufgabenstellung der Pinerkennung konnten mit dem trainierten *Faster R-CNN* Modell bereits gute Ergebnisse erzielt werden. Der zum Training und der anschließenden Evaluation genutzte Trainingsdatensatz *CHIP-EVAL* ist mit lediglich 48 Bildaufnahmen recht klein. Durch Ergänzung des Datensatzes mit weiteren Chipaufnahmen und erneutem Training ist mit einer deutlichen Steigerung der Erkennungsrate zu rechnen. Auch die entwickelte Computer Vision Pipeline lieferte gute Ergebnisse. Die untersuchten Verfahren wurden bezüglich der einstellbaren Parameter auf den eingesetzten Evaluationsdatensatz optimiert. Um Aussagen über die Praxistauglichkeit dieser Methoden treffen zu können, sind weitere Tests mit einer größeren Anzahl unterschiedlicher Chips notwendig.

Auch die untersuchten Verfahren zur Pinerkennung wurden in der Sprache Python implementiert. Sie sind zusammen mit den genutzten Metriken in einem eigenständigen Python-Modul gebündelt. Alle Verfahren implementieren dieselbe Schnittstelle und können daher je nach Anwendungsfall ausgetauscht werden.

Weiterentwicklung der Pinerkennung

Eine Verbesserung der Pinerkennung könnte vermutlich über die Einbeziehung von Metawissen über die einzelnen Chip *Packages* erreicht werden. Die Rastermaße der Pins und damit auch deren Anzahl unterliegt gewissen Normen. Über eine Validierung der Pinanzahl könnten doppelte oder auch fehlerhafte Detektionen erkannt und automatisch entfernt werden. Das Metawissen zu einem einzelnen Chip könnte etwa mit dem von Kleber et al. [9] vorgestellten Verfahren zur Dokumentenanalyse gewonnen werden. Hierfür wird eine Methodik zum automatisierten Auslesen der Chip-ID vorausgesetzt, welches ebenfalls von Kleber et al. beschrieben wird.

Die entworfene Computer Vision Pipeline beruht sehr stark auf Annahmen über die geometrische Struktur der Chips. Dieses Verfahren würde daher im Besonderen von zusätzlichem Metawissen über den zu analysierenden Chip profitieren. Bereits eine Klassifikation entsprechend der Chip *Packages* könnte dieses Verfahren stark verbessern. So könnten mit diesem Wissen etwa speziell angepasste Pipelines für die einzelnen Chip *Packages* konstruiert werden. Bei DFP-*Packages* könnte dabei etwa die Suche nach Komponentenpins auf die beiden langen Seiten des Chips beschränkt werden. Die Anzahl falsch detekterter Pins könnte damit reduziert und die *Precision* erhöht werden.

Es wäre auch die Entwicklung einer neuen Architektur eines Neuronalen Netzes vorstellbar, die sowohl die Aufgabe der Chiperkennung, als auch die Erkennung von Pins in einem einzigen Modelldurchlauf ausführt. Die Pinerkennung könnte dabei von den extrahierten Features des gesamten PCBs und Informationen über die Chip-Bounding Boxen profitieren. Dazu könnte etwa die Pinerkennung in einem separaten, abgezweigten Netzwerk ähnlich der Architektur des *Mask R-CNN* Modells [43] erfolgen. Dieses könnte dann auf die Lokalisierung der optimalen Kontaktierpunkte trainiert werden und als Output einen Vektor fixer Länge, bestehend aus Pinkoordinaten, für jeden erkannten Chip liefern.

Eine allgemeine Ergänzung zu der Analyse einzelner PCBs wäre die Wiedererkennung von bereits analysierten Platinen. Damit könnten wiederkehrende Tests, zum Beispiel bei neuen Firmware-Versionen, beschleunigt werden. Die Ergebnisse früherer Tests könnten genutzt werden, um Schwachstellen gezielter zu suchen und den Analyseprozess effizienter zu gestalten. Zur Umsetzung könnten etwa auf Merkmals- bzw. Feature-Punkten basierende Verfahren genutzt werden, wie sie etwa von Pramerdorfer und Kampel zur Erkennung und Klassifizierung von PCBs in der Abfallentsorgung genutzt werden [33].

3. Hardwareplattform für die automatisierte PCB-Kontaktierung

Bei der IT-Sicherheitsanalyse eines eingebetteten Systems wird sehr häufig mit einzelnen Komponenten über eine direkte elektrische Kontaktierung interagiert. Dazu werden einzelne Komponentenpins entweder über Federkontakte stifte temporär oder über angelötete Kabel permanent zugänglich gemacht. Die manuelle Platzierung der Federkontakte durch einen Sicherheitsanalysten setzt sehr viel Feingefühl und eine ruhige Hand voraus. Durch ein Abrutschen der Stifte, während das System unter Spannung steht, kann es zu Beschädigungen und im schlimmsten Fall zu einer vollständigen Zerstörung der zu untersuchenden Einheit kommen. Werden dieselben Pins für mehrere Analysen benötigt, werden an diese meist Steckverbindungen angelötet. Über sie sind bestimmte Pin-Kombinationen während des gesamten Analysezeitraums sehr einfach erreichbar.

Die genannten Schritte zur elektrischen Kontaktierung entsprechen dem aktuell eingesetzten Vorgehen bei der *SCHUTZWERK GmbH* und werden bisher händisch durchgeführt¹. Sie sind daher sehr zeit- und kostenaufwändig und können nur von erfahrenen Spezialisten durchgeführt werden. Die sichere Lokalisierung einzelner Pins ist bei Chips mit sehr vielen Pins ebenfalls fehleranfällig und erfordert höchste Konzentration des Sicherheitsanalysten. Ein Fehler hierbei kann ebenfalls zur Beschädigung des Untersuchungsobjekts führen.

Für die automatisierte, elektrische Kontaktierung wurde daher in dieser Arbeit eine mechanische Plattform entwickelt, welche Gegenstand dieses Kapitels ist. Zunächst werden in Kapitel 3.1 aktuell angewendete Techniken und Geräte für die elektrische Kontaktierung von PCBs vorgestellt, wie sie etwa beim *Reverse Engineering* oder auch in der Fertigung von PCBs eingesetzt werden. Kapitel 3.2 fasst die Anforderungen an eine automatische Kontaktierhardware aus Sicht der Sicherheitsanalyse eingebetteter Systeme zusammen. Ein solches System soll dabei neben der elektrischen Kontaktierung auch in der Lage sein, die für eine visuelle Analyse benötigten Bilddaten zu generieren. In dem Kapitel werden daher sowohl mechanische Anforderungen, als auch Anforderungen an ein entsprechendes Kamerasystem vorgestellt. Mögliche Lösungsansätze werden in Kapitel 3.3 diskutiert und die Entscheidung für das letztlich ausgewählte System begründet. Gegenstand von Kapitel 3.4 ist der in Form einer selbst entwickelten Hardwareplattform umgesetzte Prototyp einer Kontaktierhardware. Dabei werden die wichtigsten Designentscheidungen, zusammen mit der Auslegung der Kernkomponenten

¹Ergebnis eines Interviews mit Sicherheitsanalysten für den Bereich *Embedded Systems* der *SCHUTZWERK GmbH* am 05.03.2019

des Antriebssystems vorgestellt. Es wird weiterhin das eingesetzte Kamerasystem und die softwareseitige Ansteuerung der entwickelten Plattform beschrieben. Dabei wird insbesondere auf die verwendeten Protokolle und beteiligten Hard- und Softwarekomponenten eingegangen. Die Genauigkeit der entwickelten Plattform wird maßgeblich über die initiale Kalibrierung bestimmt. Da diese für das entwickelte System recht komplex ist, wurde diesem Teil der Arbeit ein eigenständiges Kapitel gewidmet. Die finale Evaluation der Hardwareplattform findet sich daher erst am Ende des nachfolgenden Kapitels in 4.6. Gewonnene Erkenntnisse aus der Montage und ersten Tests des Hardwareprototyps finden sich dennoch bereits in Kapitel 3.4.6. Einen abschließenden Ausblick und Anregungen für künftige Erweiterungen werden in Kapitel 3.5 gegeben.

3.1. Stand der Technik

Die automatisierte, elektrische Kontaktierung eines PCBs wird bereits seit den 1970er Jahren zum Zweck der Qualitätssicherung in der Elektronikfertigung eingesetzt. Bei dem strukturellen Testverfahren des *In-Circuit* Tests können einzelne Bauteile auf ihre korrekte Funktionsweise und Montage hin untersucht werden [5, S. 135-169]. Dazu wird mittels eines elektromechanischen Teils die elektrische Verbindung zwischen Testpunkten auf dem PCB und eingesetzter Messtechnik hergestellt. Für große Stückzahlen werden hierfür häufig Nadelbettadapter, auch *Bed of Nails* Tester genannt, eingesetzt. Die elektrische Kontaktierung erfolgt über Prüfnadeln. In einem Nadelbett sind dazu an den Stellen, an denen sich Prüfpunkte auf der zu untersuchenden Leiterplatte befinden, Bohrungen eingebracht. Diese dienen der Aufnahme von Kontaktsteckhülsen, welche ihrerseits federnd gelagerte Prüfnadeln aufnehmen. Da für jeden zu testenden Platinentyp ein eigens konstruiertes Nadelbett hergestellt werden muss, ist dieses Verfahren sehr unflexibel. Zur wirtschaftlichen Prüfung von Kleinserien und Prototypen wurde daher in den 1980er Jahren das sogenannte *Flying-Probe*-Testverfahren entwickelt [5, S. 171-191]. Die elektrische Kontaktierung erfolgt dabei mit Prüfnadeln, welche auf einem beweglichen X-Y-Z-Achssystem montiert sind. Von diesen Einheiten sind normalerweise vier für eine Seite des PCBs vorhanden. Den schematischen Aufbau eines solchen *Flying-Probe* Testers zeigt Abbildung 3.1. *Flying-Probe* Tester verfügen über eine Positioniergenauigkeit von $\pm 100 \mu\text{m}$, womit Testpunkte mit einem Durchmesser von $200 \mu\text{m}$ zuverlässig kontaktiert werden können. Zur Einmessung der zu prüfenden Platine dient eine auf einer der fahrenden Einheiten montierte Kamera und auf dem PCB angebrachte Referenzmarken. Das Kamerasystem kann auch für einfache optische Tests, wie die Überprüfung auf das Vorhandensein eines Bauteils, genutzt werden. Durch die mechanische Positionierung der Prüfnadeln ist die Testgeschwindigkeit dieser Systeme im Vergleich zu Nadelbetttestern relativ gering.

Ng gibt in seiner Arbeit einen Einblick in die Einsatzmöglichkeiten eines *Flying-Probe*-Testers beim *Reverse Engineering* von PCBs [3, S. 134-142]. Dabei betont Ng die Wichtigkeit der Software-Tools, die kompatibel mit der eingesetzten *Flying-Probe* Hardware sein müssen und

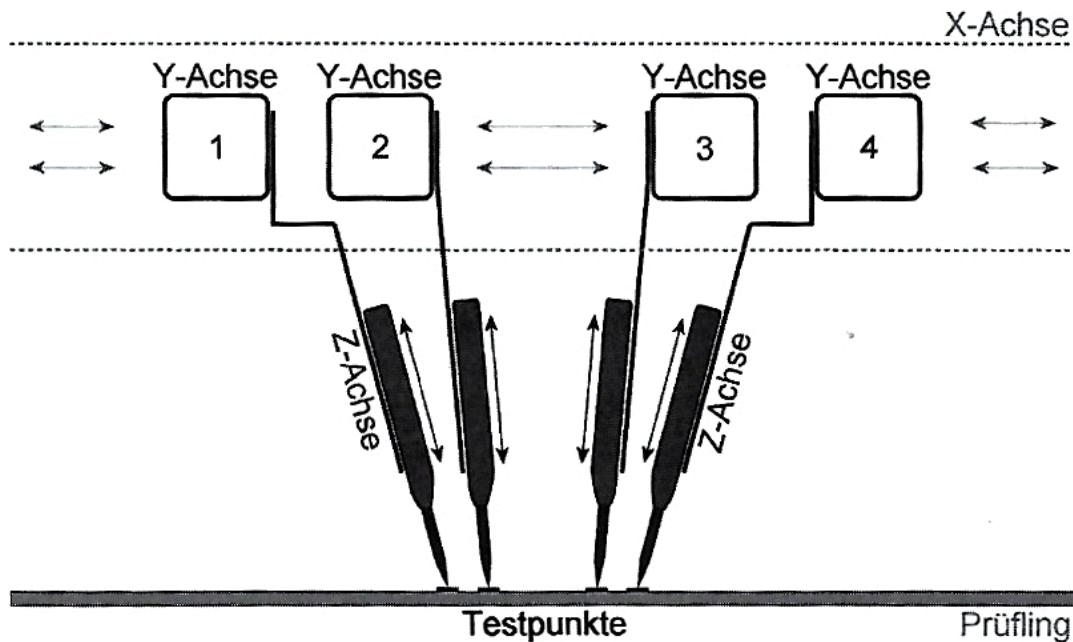


Abbildung 3.1.: Schematischer Aufbau eines *Flying-Probe* Testers [5, S. 183]

meistens von den Herstellern dieser System angeboten werden. Er beschreibt weiterhin den *Flying Probe Tester - Reverse Engineering* (FPT-RE) Prozess. Dieser besteht nach Ng im ersten Schritt aus der Erstellung einer Bildaufnahme der zu untersuchenden Platine, gefolgt von einer Identifizierung der Komponenten und Lokalisierung der Prüfpunkte. Er geht dabei ebenfalls auf aktuell verfügbare Software-Tools bekannter Hersteller von *Flying-Probe-Testers* (FPTs) ein. Der Vorgang zur Rekonstruktion mittels eines FPTs wird beschrieben und Tools zum Export der gewonnenen CAD-Daten vorgestellt.

Carlson und Hauptmann beschreiben ebenfalls den Einsatz eines FPT für das *Reverse Engineering* und den Test von PCBs, für die keine Dokumentationen mehr vorliegen [44]. Sie gehen auf den Einsatz des in Maschinen diesen Typs üblicherweise vorhandene Kamerasytems ein, um die Koordinaten von Testpunkten zu bestimmen. Auch die anschließende Rekonstruktion der Netzlisten mittels elektrischer Kontaktierung durch den Tester wird erläutert.

3.2. Anforderungen

Im Rahmen der Arbeit wurde zunächst eine Anforderungsanalyse durchgeführt, bei welcher die wichtigsten Eigenschaften eines Systems zur elektrischen Kontaktierung für die Sicherheitsanalyse eingebetteter Systeme bestimmt wurden. Dazu wurden Interviews mit Sicherheitsanalysten der *SCHUTZWERK GmbH* aus dem Bereich *Embedded Systems* geführt. Die Ergebnisse dieser Analyse sind in diesem Kapitel zusammengefasst. Dabei wurde zwischen Anforderungen an die mechanische Konstruktion zur elektrischen Kontaktierung (Kapitel 3.2.1) und den Anforderungen an ein Kamerasytem für die visuelle PCB-Analyse (Kapitel 3.2.2) unterschieden. Die

so ermittelten Anforderungen bilden die Entscheidungsgrundlage für den später umgesetzten Prototypen.

3.2.1. Mechanische Anforderungen

Zur elektrischen Kontaktierung von PCBs sollen die bereits bei FPT eingesetzten gefederten Prüfspitzen genutzt werden. Diese bieten durch den Anpressdruck der Federn eine sichere, elektrische Kontaktierung und können gleichzeitig kleine Höhenunterschiede der Bauteilpins oder anderen Kontaktierpunkten ausgleichen.

Um Tests an Debugging-Schnittstellen, wie etwa JTAG oder Bussystemen wie I²C oder SPI zu ermöglichen, müssen durch das zu entwickelnde System bis zu vier dieser Prüfspitzen pro Seite des PCBs im Raum platziert werden. Kollisionen der Einheiten sind zu vermeiden. Gleichzeitig soll jedoch möglichst viel Spielraum bei der Platzierung der Spalten vorhanden sein. In einem ersten Prototyp soll ein System mit lediglich zwei dieser Einheiten für eine einseitige Kontaktierung als *Proof of Concept* aufgebaut werden. Damit lassen sich bereits einfache Bussysteme, wie etwa I²C, oder auch elektrische Tests etwa zur Überprüfung einer elektrischen Verbindung zwischen zwei Kontaktierpunkten durchführen.

Das zu entwickelnde System muss in der Lage sein, PCBs mit einer Größe von bis zu 300 × 200 mm zu analysieren. Dies entspricht etwa den doppelten Abmessungen eines PCBs im nach DIN IEC 60297 genormten Euroformat. Karten dieser Bauart besitzen üblicherweise eine Tiefe von 160 mm und eine Höhe von 100 mm. Die maximale Dicke dieser Karten ist nicht genormt. Aus Erfahrungswerten der Mitarbeiter der *SCHUTZWERK GmbH* wurde die maximale Dicke eines vollständig bestückten PCBs mit 30 mm festgelegt. Die meisten PCBs sind jedoch flacher. Mit der gewählten Maximaldicke sollen auch große Kondensatoren berücksichtigt werden, die für das zu entwickelnde System kein Hindernis darstellen dürfen. Das System soll weiterhin in der Lage sein, sich 20 mm oberhalb der maximalen Platinendicke in einen kollisionsfreien Sicherheitsbereich zurückzuziehen. Die Prüfspitzen müssen also insgesamt einen Fahrweg von 50 mm aufweisen.

Es sollen Chips mit unterschiedlichen *Packages* kontaktiert werden können. Gängige Chips im TQFP *Package* stellen dabei mit einem Rasterabstand von 0,5 mm und einer Padbreite von 0,17 bis 0,27 mm² die kleinsten Strukturen da, die von dem neu zu entwickelnden System noch zuverlässig kontaktiert werden sollen. Daraus lässt sich die Forderung nach einer absoluten Positioniergenauigkeit von ± 0,085 mm in der Ebene ableiten. Bei der Dimensionierung und Auslegung der mechanischen Antriebselemente soll eine theoretische Genauigkeit von 0,01 mm angestrebt werden, so dass der Fehler aus diesen Elementen praktisch vernachlässigt werden kann. Die Genauigkeit des Gesamtsystems wird daher primär durch Toleranzen der restlichen mechanischen Bauteile und die Montage bestimmt.

Die Geschwindigkeit ist bei der automatisierten, elektrischen Kontaktierung nicht primär ausschlaggebend. Um dennoch Analysen in einem passablen Zeitrahmen zu ermöglichen, soll

²<http://ww1.microchip.com/downloads/en/PackagingSpec/00049w.pdf>

das System eine maximale Verfahrgeschwindigkeit von $100 \text{ mm/s} = 6 \text{ m/min}$ in der Ebene aufweisen.

Das komplett montierte System soll noch mobil sein und von einer Person bewegt werden können. Das Gesamtgewicht darf daher 15 kg nicht überschreiten. Dadurch ist sichergestellt, dass mit dem System auch Tests direkt bei Herstellern von PCBs durchgeführt werden können, die aus Sicherheitsgründen oder Auflagen die zu testende Platine nicht außer Haus geben wollen oder können.

Für den ersten Prototypen wurde ein Kostenrahmen von etwa 1000 € angesetzt. Darin enthalten sind der mechanische Grundaufbau sowie zwei frei im Raum platzierbare Prüfspitzen. Das System soll modular und gut skalierbar sein, sodass weitere Prüfspitzen ohne großen mechanischen und finanziellen Aufwand später ergänzt werden können.

In manchen Fällen kann ein PCB nicht vollständig von seinem Gehäuse isoliert werden. Als Beispiel hierfür sind etwa Steuergeräte im Automobilbereich zu nennen, bei denen die PCBs oftmals eingeklebt und mechanisch gesichert sind, um auch starken Umwelteinflüssen, wie etwa Schwingungen oder Stoßbelastungen, stand zu halten. Die Aufnahme der PCBs des Systems soll so gestaltet werden, dass auch in einem solchen Fall eine Analyse der Platine möglich ist. Dabei ist mit einem Maximalgewicht eines solchen, teilweise demontierten Prüflings von 1 kg zu rechnen.

3.2.2. Anforderungen an das Kamerasytem

Die Aufgabe des Kamerasytems ist die Erzeugung hochqualitativer Bildaufnahmen für die visuelle PCB-Analyse. Es muss möglich sein, beliebige Bildkoordinaten in Koordinaten der fahrbaren, die Prüfspitzen tragenden Einheiten umzurechnen. Angelehnt an die von Berger [5, S. 55-64] beschriebenen Anforderungen an ein Kamerasytem zur optischen Inspektion in der Elektronikfertigung wurden auch für das zu entwickelnde System die wichtigsten Parameter hinsichtlich des bildgebenden Systems festgelegt:

Da wie in Kapitel 3.2.1 beschrieben TQFP Packages mit einer minimalen Pad- bzw. Pinbreite von 0,17 mm sicher kontaktiert werden sollen, muss das Kamerasytem gemäß dem Abtasttheorem eine doppelt so hohe, minimale Pixelauflösung von 0,085 mm/Pixel aufweisen. Sie ist damit identisch zur Positioniergenauigkeit der fahrbaren Einheiten (vgl. Kapitel 3.2). Die Bildauflösung der Kamera soll maximal 1920×1080 Pixel betragen, da die damit entstehende Datenmenge noch auf Standardhardware in Echtzeit verarbeitet werden kann. Damit ergibt sich ein theoretisches maximales Field of View (FOV) von $163,2 \times 91,8$ mm.

Das Kamerasytem muss sich außerhalb des Verfahrbereichs der Prüfspitzen befinden. Es wurde daher unter Berücksichtigung eines zusätzlichen Sicherheitsabstands der selben Höhe des Verfahrbereichs der Einheiten ein Arbeitsabstand zum aufzunehmenden PCB von 100 mm festgelegt. Um auch Strukturen in unterschiedlichen Höhen, wie etwa Beschriftungen auf dem *Silkscreen* und aufgedruckte Chip-IDs, in einer Aufnahme scharf abzubilden, sollte das Kamerasytem über einen möglichst großen Tiefenschärfebereich verfügen.

Die eingesetzte Optik sollte eine sehr geringe Linsenverzeichnung aufweisen. Unter Linsenverzeichnung ist hierbei die verzerrte Darstellung eines Objekts aufgrund des Strahlengangs durch die Optik zu verstehen. Eine vorhandene Verzerrung kann jedoch auch mittels Softwarealgorithmen im Zuge der in Kapitel 4.5 vorgestellten, Kamerakalibration entfernt werden.

Um möglichst gute, reproduzierbare Bildaufnahmen zu erhalten, soll der Arbeitsbereich möglichst gleichmäßig mit einer diffusen Lichtquelle ausgeleuchtet werden. Störende Schattenwürfe hoher Bauteile sollen dadurch minimiert werden.

3.3. Ansatz

In diesem Kapitel wird der Prozess zur Auswahl einer geeigneten Plattform für die automatisierte elektrische PCB-Kontaktierung beschrieben. Dazu werden zunächst in Kapitel 3.3.1 bereits verfügbare elektromechanische Plattformen vorgestellt, die für diesen Zweck genutzt werden können. Dem Aufbau verschiedener Varianten von Kamerasyystemen zur Erzeugung von Bildaufnahmen, welche zur visuellen Analyse eines PCBs benötigt werden, widmet sich Kapitel 3.3.2. Der Auswahlprozess und die dabei getroffene Wahl für die jeweiligen Teilsysteme sind Gegenstand von Kapitel 3.3.3.

3.3.1. Mögliche mechanische Plattformen zur PCB-Kontaktierung

Für das elektromechanische System zur Kontaktierung von PCBs wurden vier mögliche, mechanische Plattformen betrachtet.

Flying-Probe Tester

Ein naheliegender Ansatz ist die Verwendung eines FPT, wie er in Abbildung 3.2a abgebildet ist. Diese hochpräzisen und zugleich hochproduktiven Systeme sind in der Lage mehrere hundert Kontaktierungen pro Sekunde bei Genauigkeiten im Bereich von wenigen hundertstel Millimetern durchzuführen³. Dabei werden hochgenaue und sehr dynamische Linearmotoren zusammen mit optischen Encodern zur exakten Positionierung der Prüfspitzen eingesetzt. Zur Ansteuerung werden von den Herstellern meist proprietäre Softwaresysteme angeboten. Marktübliche Systeme besitzen bis zu vier Prüfspitzen pro Seite. Der Kaufpreis solcher Systeme liegt im Bereich von mehreren hunderttausend Euro⁴.

3D-Drucker

Ein deutlich preiswerterer Ansatz ist die Nutzung eines 3D-Drucker Bausatzes als Ausgangspunkt für einen mechanischen Aufbau zur PCB-Kontaktierung. Ein Beispiel eines solchen 3D-Druckers zeigt Abbildung 3.2b. Bausätze dieser Art sind recht kostengünstig mit Preisen im Bereich

³<https://www.spea-ate.de/index.php?id=412>

⁴Telfonat am 12.04.2019 mit Herrn Hubertus, SPEA GmbH



Abbildung 3.2.: Mögliche mechanische Plattformen eines elektrischen Kontaktiersystems

von mehreren hundert Euro. Aufgrund der stetig wachsenden Verbreitung sind diese gut erprobt und werden laufend durch eine wachsende Community verbessert. Die erzielbare Positioniergenauigkeit liegt je nach Modell im Bereich von mehreren zehntel bis zu einigen hundertstel Millimetern. Dabei kommen als Antriebe überwiegend kostengünstige Schrittmotoren in Kombination mit Riemenantrieben oder Trapezgewindespindeln ohne Positionsrückmeldung zum Einsatz. Zur Ansteuerung wird oftmals OpenSource Software verwendet. Ein bekanntes Projekt ist etwa das auf der Arduino Plattform basierende, auf GitHub veröffentlichte Projekt *Marlin*⁵. Allerdings sind 3D-Drucker meist nur für einen einzelnen, frei im Raum positionierbaren Druckkopf ausgelegt.

Industrieroboter

Auch Industrieroboter könnten für die elektrische Kontaktierung eingesetzt werden. Zu Ausbildungszwecken und dem Einsatz in der Lehre etwa sind kleine, einfache Modelle bereits für einige hundert Euro erhältlich. Abbildung 3.2c zeigt etwa den Roboterarm *uArm Swift Pro* der Firma *UFACTORY*. Bei einem Preis von ca. 750 \$⁶ liefert dieses System eine Wiederholgenauigkeit von 0,2 mm sowie eine Geschwindigkeit von 20 mm/s. Durch die gewählte Kinematik ergibt sich ein nicht rechtwinkliger Arbeitsbereich in einem Bereich von 50 bis 320 mm. Zur Ansteuerung des Systems werden Softwareschnittstellen in gängigen Programmiersprachen, wie etwa Python angeboten.

Neukonstruktion

Neben den bisher genannten, bereits existierenden Lösungen ist auch eine vollständige Neukonstruktion einer Hardwareplattform vorstellbar. Diese könnte exakt auf die Bedürfnisse der Sicherheitsanalyse eingebetteter System zugeschnitten und entsprechend entworfen werden.

⁵<https://github.com/MarlinFirmware/Marlin>

⁶<https://store.ufactory.cc/products/uarm>, Stand 28.08.2019, 12:00 Uhr

Obgleich dieser Ansatz den größtmöglichen Gestaltungsspielraum bietet, bringt er gleichzeitig auch den größten Arbeits- und Zeitaufwand mit sich. Neben der reinen Konstruktion und dem Design müssen Einzelteile bestellt und das System montiert werden. Das Gesamtsystem muss evaluiert und auf seine Tauglichkeit für die Aufgabenstellung hin überprüft werden. Erfahrungsgemäß ist dies ein iterativer Prozess, bei dem häufig Änderungen an der Konstruktion nach dem Zusammenbau und der Durchführung erster Tests vorgenommen werden müssen.

3.3.2. Mögliche Varianten eines Kamerasytems

Für ein Kamerasytem zur Generierung von PCB-Bildaufnahmen sind zwei unterschiedliche Ansätze vorstellbar.

Statisches Kamerasytem

Zum einen könnte ein Kamerasytem mit feststehender Kamera eingesetzt werden. Dazu wird etwa eine hochauflösende Digitalkamera fest am Rahmen der mechanischen Kontaktierhardware oder auf einem separaten Stativ angebracht. Zu Beginn des visuellen Scanvorgangs fahren alle Kontaktiereinheiten zur Seite, so dass das Kamerasytem freien Blick auf die zu untersuchende Platine erhält. Anschließend können mehrere Bildaufnahmen zum Beispiel mit unterschiedlichen Beleuchtungen aufgenommen werden, um verschiedene Details des PCBs gezielt zu betonen. Die Auswertung der Aufnahmen erfolgt offline auf einem separaten Rechner. Nach erfolgreicher Analyse können die Kontaktiereinheiten wieder in den Arbeitsbereich einfahren und die Prüfspitzen für die durchzuführenden weiteren Analysen platzieren.

In einer Aufnahme des Kamerabildes muss somit das gesamte PCB erfasst werden. Das Kamerasytem wird dazu so eingerichtet, dass ein PCB mit maximaler Größe vollständig abgelichtet werden kann. Bei kleineren Platinen mit unter Umständen auch kompakteren Komponenten entsteht dann das Problem, dass durch den prinzipbedingt groß gewählten Bildausschnitt kleinere Details durch die begrenzte Kameraauflösung nur noch unzureichend aufgelöst werden. Weiterhin wird das PCB während der elektrischen Kontaktierung durch die fahrenden Einheiten verdeckt. Ein visuelles Feedback durch das Kamerasytem oder die Erzeugung von Bildaufnahmen zum Zwecke der Dokumentation ist nicht möglich. Zur Kontrolle der Lichtverhältnisse muss ein abgeschirmter Bereich, ähnlich dem von Kleber et al. [9] beschriebenen Aufbau, geschaffen werden. Ein solcher Aufbau könnte die fahrbaren Einheiten in ihrer Bewegungsfreiheit einschränken.

Fahrbares Kamerasytem

Alternativ zu einer feststehenden Kamera könnte auch ein fahrbares Kamerasytem genutzt werden. Nach Berger [5, S. 176] kommen solche Systeme auch in FPTs zur Bestimmung der Ausrichtung der zu testenden PCBs zum Einsatz. Der in Kapitel 3.3.1 vorgestellte Roboterarm *uArm Swift Pro* verfügt ebenfalls bereits über eine integrierte Kamerahalterung.

Da sich die Kamera bei einem fahrenden Kamerasystem meist relativ nah an dem aufzunehmenden Objekt befindet, könnte ein preiswertes USB-Mikroskop an dieser Stelle als bildgebendes System verwendet werden. Diese meist sehr kompakten, zylinderförmigen Kameras verfügen über ein Makroobjektiv, das auch die Aufnahme von sehr kleinen Strukturen ermöglicht. Der in einer Bildaufnahme sichtbare Bereich ist dabei jedoch sehr klein. Gesamtaufnahmen eines PCBs müssen daher durch Zusammenfügen mehrerer Einzelaufnahmen generiert werden. Dieser je nach eingesetztem Algorithmus sehr rechenintensive Prozess wird auch als *Image Stitching* bezeichnet [48, S. 377 f.]. Er liefert als Ergebnis ein sehr detailliertes Bild, dessen Auflösung um den Faktor der Anzahl der Einzelaufnahmen größer sein kann.

Durch die Tatsache, dass sich die Kamera im Verfahrbereich der Prüfeinheiten befindet, lässt sich die Beleuchtung etwa durch eine lichtundurchlässige Kompletteneinhausung des Gesamtsystems sehr einfach kontrollieren. Dazu können ansteuerbare Lichtquellen, wie zum Beispiel LED-Streifen, im Arbeitsraum angebracht werden. Die Farbe und Intensität kann dann softwaretechnisch für die Bildaufnahmen angepasst werden.

Eine auf einer fahrbaren Prüfeinheit montierte Kamera bedeutet jedoch auch, dass diese Einheiten mehr Platz benötigen. Dies kann zu einem insgesamt größeren Gesamtsystem führen. Je nach Wahl der Stelle, an welcher die Kamera an die fahrende Einheit montiert wird, müssen zusätzlich größere Verfahrwege eingeplant und Bauteile größer konstruiert werden. Letzteres kann durch größere mechanische Hebel in ungünstigeren Kraftverhältnissen resultieren, welche sich wiederum negativ auf die Antrieb und die Maschinengenauigkeit auswirken können.

3.3.3. Auswahl der Teilsysteme

Aufgrund des hohen Anschaffungspreises und der meist geschlossenen Systemarchitektur wurde der Einsatz eines FPT für die elektrische Kontaktierung im Rahmen dieser Arbeit ausgeschlossen. Die hohen Verfahrgeschwindigkeiten spielen bei der Sicherheitsanalyse von PCBs, im Gegensatz zu der Qualitätssicherung in der Elektronikfertigung, nur eine untergeordnete Rolle. Der hohe Preis für die eingesetzten Antriebstechnologien in diesen Systemen ist daher an dieser Stelle nicht zu rechtfertigen. Auch die Geschlossenheit der Steuerungssysteme ist von großem Nachteil. Angepasste Prüf- und Analysezyklen ließen sich nur in Zusammenarbeit mit den Herstellern entwickeln.

Die Idee, einen Roboterarm für die elektrische Kontaktierung einzusetzen, wurde ebenfalls verworfen. Gründe hierfür waren vor allem die mechanischen Eigenschaften des in Kapitel 3.3.1 ausgesuchten Systems: Eine mit 0,2 mm angegebene Wiederholgenauigkeit ist zur Kontaktierung eines Chips mit TQFP Package bei einer minimalen Padbreite von 0,17 mm nicht ausreichend. Durch die gewählte Kinematik mit langen, dünnen Hebeln, neigt dieser Aufbau außerdem bei ruckartigen Bewegungen zu Schwingungen.

Um einen 3D-Drucker zur Platzierung von mehreren Prüfspitzen auf einem PCB einzusetzen, müsste dieser grundlegend modifiziert werden. Bei den meisten dieser Maschinen besitzt der Druckkopf selbst nur zwei Freiheitsgrade und kann sich damit in einer Ebene bewegen. Die

dritte Achse befindet sich im Druckbett. Der Aufbau eines Systems mit mehreren Einheiten, die vollständig autonom verfahren werden können, ist damit nicht möglich.

Daher wurde beschlossen, eine eigens für den Zweck der Sicherheitsanalyse eingebetteter Systeme designte Hard- und Softwareplattform zu entwickeln. Die entwickelte Plattform soll später im Sinne des *OpenHardware*-Gedankens kostenlos, öffentlich verfügbar gestellt werden. Zur Konstruktion soll daher bevorzugt *OpenSource*-Software, wie etwa das kostenlose Computer Aided Design (CAD)-Programm *FreeCAD*⁷, eingesetzt werden, um Anpassungen und Verbesserungen durch andere Nutzer zu fördern. Die Konstruktion soll sich dabei an die bereits beschriebenen 3D-Drucker Bausätze anlehnen. Darin verwendete Bauteile, wie etwa die Antriebs- und Steuerelemente, sind durch die große Nachfrage nach 3D-Druckern sehr kostengünstig zu erwerben. Für den mechanischen Grundaufbau sollen überwiegend Aluminiumprofile genutzt werden. Diese sind ebenfalls als Standardware zu guten Konditionen auch bereits in kleinen Stückzahlen günstig zu kaufen. Das Gesamtsystem soll möglichst kompakt und modular konstruiert werden. In einem ersten Prototypen sollen zunächst zwei fahrbare Prüfeinheiten für eine einseitige Kontaktierung aufgebaut werden. Eine Erweiterung auf bis zu vier Einheiten sowie eine beidseitige Kontaktierung von PCBs sollen jedoch bereits bei der Konstruktion berücksichtigt werden. Die Aufnahmen der Prüfspitzen sollen möglichst einfach, ohne großen Montageaufwand zu wechseln sein. Somit können ebenfalls angepasste, für die jeweiligen Analysen optimierte Aufnahmen eingesetzt werden. Als Beispiel sei an dieser Stelle etwa die Aufnahme eines Oszilloskop-Tastkopfs zu nennen, mit dem Signale aufgrund der angepassten Impedanz besser abgegriffen werden können.

Die Wahl des Kamerasystems fiel auf eine fahrende Kamera mit dem in Kapitel 3.3.2 erwähnten USB-Mikroskop. Durch die Möglichkeit zur Erzeugung sehr hoch aufgelöster Aufnahmen von theoretisch beliebig großen PCBs, bietet diese Technik im Hinblick auf immer kleiner werdende Komponenten das größte Potential. Da der elektromechanische Aufbau zur PCB-Kontaktierung von Grund auf neu konstruiert wird, lässt sich ein solches System von Beginn an sehr gut in das Design integrieren. Die ausgewählte Kamera in Form eines USB-Mikroskops ist zudem sehr kostengünstig.

3.4. Entwickelte Hardware-Plattform

In diesem Kapitel wird die im Rahmen dieser Arbeit neu entwickelte Hardwareplattform zur elektrischen Kontaktierung von PCBs vorgestellt. Zunächst wird in Kapitel 3.4.1 der grundlegende mechanische Aufbau und die gewählte Kinematik beschrieben. Die Auswahl und Auslegung der mechanischen Komponenten mit dem größten Einfluss auf die Positioniergenauigkeit und die Geschwindigkeit des Systems, die Führungs- und Antriebselemente, finden sich in Kapitel 3.4.2 und 3.4.3. Einen Überblick über die Ansteuerung der Hardwareplattform und die dazu eingesetzten Hard- und Softwarekomponenten gibt Kapitel 3.4.4. Der Umsetzung des

⁷<https://www.freecadweb.org/>

Entwurfs in Form eines Prototyps mit zwei Prüfspitzen widmet sich Kapitel 3.4.5. Erkenntnisse, die bei der Montage und ersten Tests mit dem Prototypen gewonnen wurden, fasst abschließend Kapitel 3.4.6 zusammen.

3.4.1. Kinematik

Den Begriff der Kinematik definieren Gevatter und Grünhaupt wie folgt:

„Die Kinematik ist der Teil der Mechanik, in dem allein die Bewegung der Körper ohne Rücksicht auf die sie verursachenden Kräfte untersucht wird. Dabei hat die Kinematik rein geometrischen Charakter.“ [49, S. 741]

Dabei bestehen nach Gevatter und Grünhaupt die Hauptaufgaben der Kinematik in dem „Herstellen der räumlichen Zuordnung zwischen dem Werkstück, dem Werkzeug [...] sowie das Führen des Werkzeugs“ [49, S. 743]. Sie definiert also, wie und auf welchen Bahnen ein Aktor bzw. ein Werkzeug relativ zum Werkstück bewegt werden kann. Auf die zu entwickelnde Kontaktierhardware übertragen bedeutet dies, dass durch die Kinematik festgelegt wird, wie die Prüfspitzen als Akteure innerhalb des Arbeitsraums relativ zu einem PCB als Werkstück verfahren werden können. Sie hat außerdem maßgeblich Auswirkung darauf, wie die einzelnen Verfahreinheiten zueinander bewegt werden können und ob bzw. welche geometrischen Zwangsbedingungen zwischen diesen bestehen.

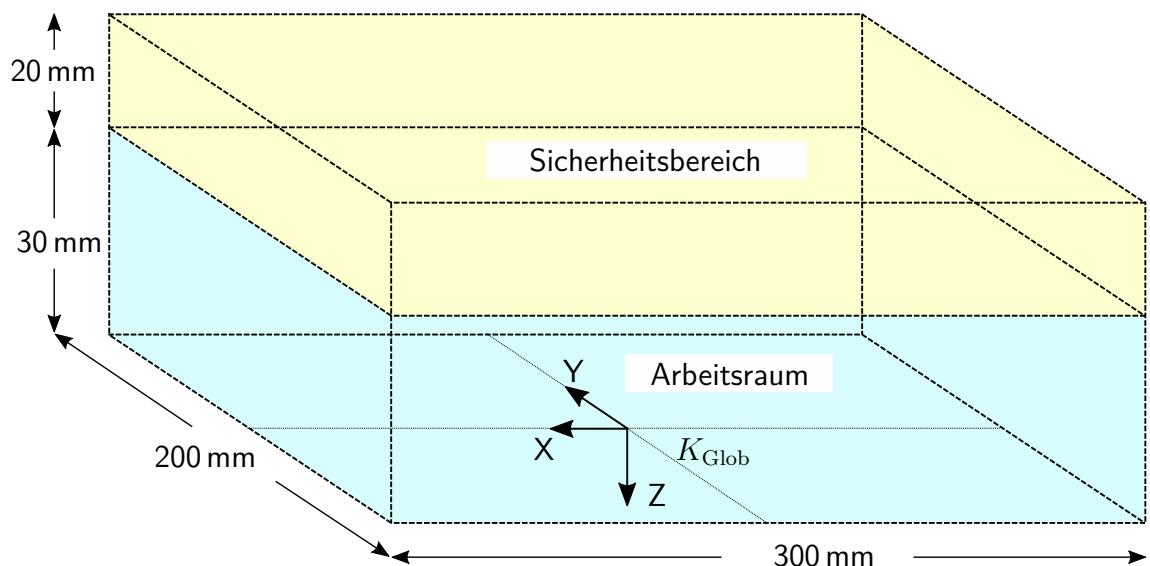
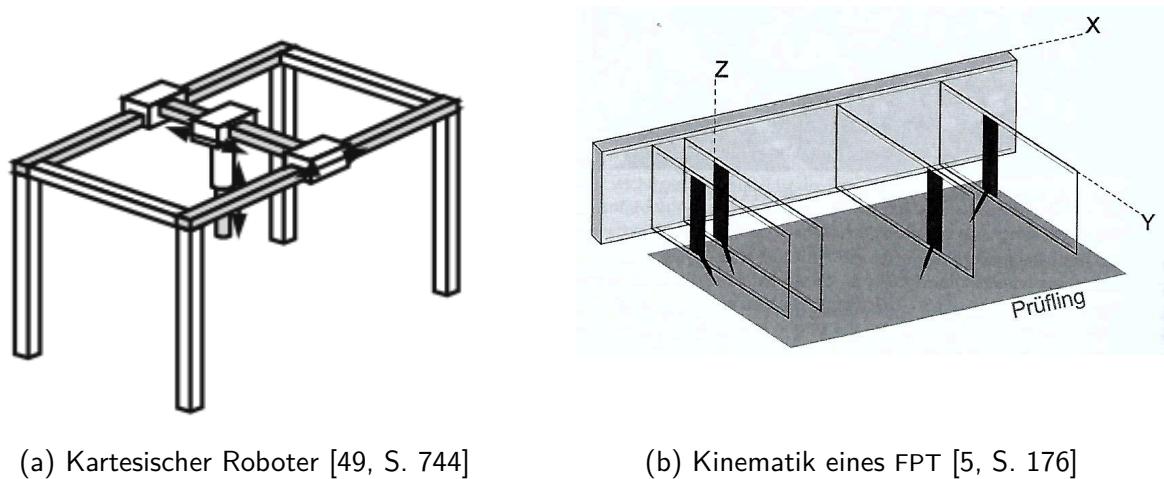


Abbildung 3.3.: Arbeitsraum und globales Koordinatensystem der Kontaktierhardware

Aus den in Kapitel 3.2 definierten maximalen PCB Abmessungen und dem ebenfalls einzuuhaltenden Sicherheitsbereich oberhalb des PCBs ergibt sich als Arbeitsraum ein Volumen mit den Abmessungen $300 \times 200 \times 50$ mm. Innerhalb dieses Volumens wurde ein virtuelles, kartesisches Koordinatensystem in dessen Zentrum definiert, das im Weiteren als globales Maschinensystem K_{Glob} bezeichnet wird. Diese Anordnung ist in Abbildung 3.3 zu sehen. Jede

Verfahreinheit muss in der Lage sein, die getragene Prüfspitze an jedem Punkt in diesem Arbeitsraum zu platzieren. Dabei besitzt jede Verfahreinheit und damit auch jede Prüfspitze drei translatorische Freiheitsgrade entlang der als X, Y und Z bezeichneten Hauptachsen des globalen Maschinensystems.

Für den zu entwickelnden Prototypen wurde aufgrund des quaderförmigen Arbeitsraums eine serielle Kinematik bestehend aus jeweils drei translatorischen Achsen pro Verfahreinheit ausgewählt. Dieser Aufbau wird nach DIN EN ISO 8373 auch als kartesischer Roboter bezeichnet und ist in Abbildung 3.4a zu sehen. Nach Berger [5, S. 176] wird diese Kinematik auch in den X-Y-Z-Achssystemen von FPTs eingesetzt. Bei FPT kommt erschwerend hinzu, dass sich mehrere Einheiten gleichzeitig im Arbeitsraum bewegen müssen. Hierzu wird nach Berger häufig eine Kinematik gewählt, bei der sich alle Einheiten eine Achse teilen, jedoch weiterhin unabhängig auf dieser bewegt werden können. Die restlichen zwei Achsen sind für jede Einheit separat ausgeführt. Eine schematische Darstellung dieses Aufbaus ist in Abbildung 3.4b und 3.1 zu sehen.



(a) Kartesischer Roboter [49, S. 744]

(b) Kinematik eines FPT [5, S. 176]

Abbildung 3.4.: Kartesische Kinematiken

3.4.2. Linearführungssysteme

Die Aufgabe eines Linearführungssystems ist die präzise Führung eines beweglichen Schlittens entlang einer vorgegebenen Achse. Dabei bestehen die Systeme meist aus einem oder mehreren beweglichen Schlitten und einer feststehenden Führungsschiene. An dieser Stelle soll nur ein kurzer Überblick, ohne Anspruch auf Vollständigkeit, über bestehende Führungssysteme für Linearbewegungen gegeben werden.

Kategorisierung von Linearführungssystemen

Führungssysteme lassen sich entsprechend der verbleibenden Freiheitsgrade kategorisieren. Führungen mit lediglich einem einzigen Freiheitsgrad erlauben ausschließlich die translatorische

Bewegung entlang einer Achse. Translatorische Freiheitsgrade in den restlichen Achsen sowie rotatorische Bewegungen werden durch die Führung unterdrückt und entsprechende Kräfte aufgenommen. Im Gegensatz dazu existieren auch Führungssysteme, die zusätzlich eine Rotation entlang der Bewegungsachse zulassen. Der bewegte Teil der Führung besitzt somit nach wie vor zwei Freiheitsgrade.

Ein weiteres Merkmal, nach welchem Linearführungen unterschieden werden können, betrifft die Gestaltung des Zwischenraums der relativ zueinander bewegten Teile. Bei Gleitlagern gleitet der bewegliche Führungsschlitten direkt auf der feststehenden Führungsschiene. Zur Verminderung der Reibung zwischen den Bauteilen werden trennende, reibungsmindernde Medien, wie etwa Schmieröl oder auch spezielle Materialpaarungen, wie etwa metallische Führungsschienen und Gleitelemente aus tribologisch optimiertem Kunststoff, eingesetzt. Im Gegensatz dazu finden sich bei sogenannten Wälzführungen kugel- oder zylinderförmige Wälzkörper im Zwischenraum zwischen Führungsschlitten und -schiene. Die Relativbewegung der Bauteile wird durch Abrollen der Wälzkörper ermöglicht. Gleitgelagerte Systeme sind oftmals ruhiger als wälzgelagerte Systeme, da sie keine beweglichen Teile enthalten. Mit Gleitelementen aus Kunststoff ausgeführten Gleitlager sind zudem nahezu wartungsfrei.

Zur Konstruktion eines möglichst kostengünstig und leicht nachzubauenden Prototyps wurde die Auswahl der Linearführungssysteme auf gängige, in 3D-Druckern zu findende Systeme, beschränkt. Diese sind in Abbildung 3.5 zu sehen.

Rundführungen

Eine sehr kostengünstige Version einer Linearführung ist die in Abbildung 3.5a dargestellte Rund- oder Linearwellenführung [50, S. 360]. Als Führungsschienen kommen dabei präzisionsgeschliffene Stangen zum Einsatz. Die beweglichen Schlitten sind häufig entweder mit Gleitelementen aus Kunststoff oder als wälzgelagerte Ausführung mit kleinen metallischen Kugeln als Wälzkörper ausgestattet. Sie erlauben dem bewegten Schlitten neben der translatorischen Bewegung auch eine Rotation um die Achse der Führungsstange. Diese Führungsstangen werden normalerweise frei schwebend montiert und nur an den Enden gelagert. Die Steifigkeit des Führungssystems wird daher maßgeblich durch die Wahl des Durchmessers der Führungsstangen und deren Länge bestimmt. Für eine überschlägige Auslegung können die Führungsstangen als einfache Biegebalken betrachtet werden.

Supported Rails

Die in Abbildung 3.5b abgebildeten *Supported Rails* genannten Führungen ähneln sehr stark den bereits vorgestellten Rundführungen. Durch eine Unterstützung der Führungsstangen auf ganzer Länge können sie aber von der Steifigkeit des Montageuntergrunds profitieren. Sie lassen eine Rotation um die Führungsachse nur in einem bestimmten Bereich zu. Durch die Stützstruktur der Führungsstangen ist dieses Führungssystem teurer als ein entsprechendes Rundführungssystem.



(a) Rundführungen [51]

(b) Supported Rails [52]

(c) Profilschienenführung [53]

Abbildung 3.5.: Ausgewählte Linearführungssysteme

Profilschienenführungen

Die unter anderem in qualitativ hochwertigen 3D-Druckern eingesetzten Profilschienenführungen sind in Abbildung 3.5c zu sehen. Bei diesem Typ eines Linearführungssystems rollen kugel- oder zylinderförmige Wälzkörper in geschliffenen Führungsbahnen der Führungsschienen. Im Gegensatz zu den bisher genannten Führungssystemen besitzen die Schlitten einer Profilschienenführung nur einen einzigen translatorischen Freiheitsgrad. Rotationen verursachende Momente werden durch die Führung aufgenommen [50, S. 359]. Systeme diese Art sind vorspannbar und besitzen damit eine sehr hohe Steifigkeit bei gleichzeitig sehr kompakter Bauform. Wie auch die *Supported Rails* profitieren die Führungsschienen durch eine flächige Montage von der Steifigkeit des Untergrunds. Aufgrund des vergleichsweise komplizierten Fertigungsprozesses handelt sich hierbei allerdings auch um die teuerste Variante der untersuchten Führungssysteme. Die eingesetzten Wälzkörper müssen regelmäßig mit Fett oder Schmieröl versorgt werden. Der Wartungsaufwand ist dementsprechend höher als beispielsweise bei Kunststoffgleitlagern.

Auswahl der Führungssysteme

Die in Kapitel 3.4.1 ausgewählte Kinematik sieht eine gemeinsame X-Achse für alle fahrenden Einheiten vor. Auf dieser aufbauend sind dann die jeweiligen Y- und Z-Achsen angeordnet. Da die X-Achse somit die Basis aller Einheiten darstellt, wurde für diese eine Profilschienenführung ausgewählt. Durch die flächige Montage bietet sie auch bei langen Verfahrwegen eine hohe Steifigkeit bei gleichzeitig minimalem Platzbedarf. Durch die Wahl zusätzlich vorgespannter Führungswagen kann die Steifigkeit weiter erhöht werden und bietet daher noch Optimierungspotential.

Auch für das letzte Glied der kinematischen Kette, der Z-Achse, fiel die Wahl auf eine Profilschienenführung. Diese ist primär durch den sehr kompakten Aufbau dieses Führungssystems begründet. Bei Rundführungen oder *Supported Rails* werden zur Vermeidung einer Rotation um die Führungsachse immer zwei Führungsschienen benötigt. Im Gegensatz dazu ist eine einzelne Profilschienenführung in der Lage, auch diesen Freiheitsgrad zu unterdrücken. Sie lässt lediglich eine rein translatorische Bewegung entlang der Führung zu. Durch eine umgekehrte Montage, bei welcher der Schlitten fest montiert und die Führungsschiene bewegt wird, lassen sich bei

einer sehr kompakten Konstruktion ohne große mechanische Hebel alle Bauteile vollständig aus dem Arbeitsraum fahren.

Die Y-Achse der fahrbaren Einheiten wurden aus Kostengründen als Rundführung ausgeführt. Wartungsfreie Gleitbuchsen aus Kunststoff ermöglichen einen ruhigen Betrieb. Im Gegensatz zur X-Achse ist der Verfahrweg in dieser Achse um 50 % mit 200 mm deutlich kürzer. Durch eine entsprechende Dimensionierung der Führungsstangen kann daher auch durch Rundführungen die benötigte Steifigkeit sichergestellt werden.

3.4.3. Antriebe

Die präzise Positionierung der Schlitten eines Linearführungssystems erfolgt durch ein entsprechend den Anforderungen an die statischen und dynamischen Eigenschaften ausgelegtes Antriebssystem. Wichtige Kenngrößen sind hierbei die erreichbare Genauigkeit, Steifigkeit und Geschwindigkeit. Auch das dynamische Verhalten, wie etwa die Neigung zu Schwingungen oder das Beschleunigungsvermögen des Systems, können dabei berücksichtigt werden. Da für das zu entwickelnde System zumindest bezüglich des Beschleunigungsvermögens keine Anforderungen gestellt wurden, beziehen sich die im Weiteren vorgestellten Auslegungen immer nur auf statische Kenngrößen.

Für lineare Antriebssysteme stehen eine Vielzahl technischer Lösungen zur Verfügung. Im Weiteren wird die Auswahl, wie auch bereits bei den Linearführungssystemen, auf kostengünstige, im 3D-Druck Bereich übliche Techniken reduziert [54]. Hightech-Lösungen, wie die in FPTs eingesetzten Lineardirektantriebe oder in hochpräzisen Werkzeugmaschinen eingesetzte vorgespannte Kugelumlaufspindeln mit Synchronservomotoren [50, S. 476], werden daher an dieser Stelle von weiteren Betrachtungen ausgeschlossen. Im Nachfolgenden werden die zwei am weitesten im Bereich von 3D-Druckern eingesetzten Antriebstechniken, der Antrieb mittels Synchronriemen und Trapezgewindespindel, vorgestellt⁸. Es werden die Vor- und Nachteile beider Techniken beschrieben und die durchgeführten Auslegungsberechnungen vorgestellt.

Beide Antriebstechniken nutzen mechanische Übertragungselemente, um eine rotatorische Bewegung in eine lineare umzuwandeln. Als kostengünstige Lösung werden hierfür in einfachen Anwendungen sogenannte Schrittmotoren eingesetzt. Diese können durch elektrische Impulse in diskreten Winkelschritten ohne Rückmeldung der aktuellen Rotorposition in Form einer offenen Steuerung betrieben werden. Wird die Motorlast entsprechend klein gewählt, kann hiermit auch ohne aufwendige Regelung eine sehr hohe Positioniergenauigkeit erzielt werden [50, S. 25-29].

Riementrieb

In 3D-Druckern und Automatisierungseinrichtungen werden für lineare Bewegungen unter anderem Synchronriemenantriebe eingesetzt. Ein fahrbarer Schlitten wird dabei mit einem Riemen aus verstärktem Kunststoff verbunden. Dieser wird wiederum, wie bei einem normalen

⁸https://reprap.org/wiki/RepRap_Options#X.2FY.2FZ_Axis_Motion

Riemenantrieb, durch ein Ritzel und einen rotierenden Motor angetrieben. Die dabei eingesetzte Riemenform gewährleistet durch eine formschlüssige Verbindung mit dem Ritzel ein spielfreies Antriebssystem [55, S. 584].

Vorteile des Riemenantriebs sind die sehr geringen Anschaffungskosten. Die benötigten mechanischen Bauteile, wie Riemen, Ritzel und die beschriebenen Schrittmotoren, sind sehr preiswert käuflich zu erwerben. Der Wartungsaufwand eines solchen Systems ist außerdem sehr gering. Lediglich die Riemenspannung muss in regelmäßigen Intervallen überprüft und bei Bedarf nachgespannt werden. Mit Riemenantrieben sind außerdem sehr hohe Verfahrgeschwindigkeiten durch die relativ große mechanische Übersetzung möglich.

Die Positioniergenauigkeit eines solchen Antriebs ist aufgrund der großen mechanischen Übersetzung und der eingeschränkten Winkelgenauigkeit der Motoren limitiert. Durch die Riemenspannung entsteht eine permanente mechanische Belastung, die von den umliegenden mechanischen Bauteilen aufgenommen werden muss. Ohne eine separate Lagerung des Ritzels kann sich die Vorspannung negativ auf die Lebensdauer des gewählten Motors auswirken, da die mechanischen Kräfte von der Motorlagerung aufgenommen werden müssen.

Eine wichtige Eigenschaft eines Synchronriemens ist dessen Zahnteilung $p \in \mathbb{R}^+$. Diese gibt den Abstand zwischen zwei Zähnen an. Zur Auslegung eines Synchronriemenantriebs ist zudem der Durchmesser des Antriebsritzels zu bestimmen, da dieser maßgeblich für die mechanische Übersetzung der rotatorischen Bewegung des Motors in die lineare Bewegung des Schlittens verantwortlich ist. Zur Angabe der Ritzelgröße wird dabei der effektive Umfang als Vielfaches $z \in \mathbb{N}^+$ der Zahnteilung p angegeben. Den Zusammenhang zwischen dem linearen Verfahrweg $l \in \mathbb{R}$ des Schlittens und dem Motorwinkel $\alpha_{Mot} \in \mathbb{R}$ in Grad zeigt folgende Gleichung

$$l = z \cdot p \cdot \frac{\alpha_{Mot}}{2\pi} \quad (3.1)$$

Der vom Schlitten zurückgelegte Weg kann dabei als Abwicklung des Riems durch den Winkel α_{Mot} auf dem effektiven Ritzeldurchmesser $d_{Ritzel} = p \cdot z$ verstanden werden. Abgeleitet aus Gleichung (3.1) kann nun die theoretische Positioniergenauigkeit l_{min} eines Synchronriemenantriebs mit einem Schrittmotor mit $s \in \mathbb{N}^+$ Schritten pro Umdrehung und einer Mikroschrittauflösung von $s_{Mikro} \in \mathbb{N}^+$ Mikroschritten pro Vollschritt berechnet werden zu

$$l_{min} = z \cdot p \cdot \frac{1}{s \cdot s_{Mikro}} \quad (3.2)$$

Weiterhin kann die maximal erreichbare lineare Schlittengeschwindigkeit $v_{max} \in \mathbb{R}^+$ in mm/s durch die maximale Rotationsgeschwindigkeit $v_{Mot,max} \in \mathbb{R}^+$ in Umdrehungen pro Minute nach folgender Gleichung berechnet werden

$$v_{max} = \frac{1}{60} \cdot z \cdot p \cdot v_{Mot,max}$$

Die Auslegung eines Riemensystems ist ein iterativer Prozess, bei dem eine Kombination aus Riemen- und Motorparametern gesucht wird, welche den gestellten Anforderungen genügt. Für

Profil	p in mm	z in mm	s	s_{Mikro}	$v_{Mot,max}$ in min^{-1}	l_{min} in mm	v_{max} in mm/s
GT2	2	16	200	16	400	0,010	213,3
HTD-3M	3	10	200	16	400	0,009	200

Tabelle 3.1.: Auslegung der Synchronriemenantriebe

den zu konstruierenden Prototypen wurden die Riemenantriebe bezüglich der geforderten Positioniergenauigkeit und der maximalen Verfahrgeschwindigkeit ausgelegt. Das benötigte Motormoment wurde als vernachlässigbar gering eingeschätzt. Da keine statischen Lasten zu erwarten sind, wird das Motormoment lediglich zur Beschleunigung der Schlitten benötigt. Bezuglich der Dynamik wurden keine Anforderungen an das System gestellt, daher erfolgte keine explizite Auslegung des Motors hinsichtlich dieser Kenngröße, sondern es wurde ein Motor mittlerer Größe ausgewählt. Bei der Antriebsauslegung wurden zwei gängige Riemenprofile, das *HTD* und *GT2* Profil⁹, untersucht. Als Antriebsmotor wurde jeweils ein NEMA 14 Schrittmotor mittlerer Größe angesetzt(vgl. Anhang C). Die Berechnungsergebnisse und Motorkenndaten finden sich in Tabelle 3.1.

Um eine möglichst kleine mechanische Übersetzung zu erhalten, wurden die kleinsten, verfügbaren Motorritzel ausgewählt (vgl. Gleichung (3.2)).Die geforderte Positioniergenauigkeit von 0,01 mm konnte dabei nur mit einem 16-fachen Mikrostepping erzielt werden. Die mit ca. 200 mm/s erreichbaren Maximalgeschwindigkeiten sind sehr gut und entsprechen dem Zweifachen des geforderten Wertes, so dass mit diesen Antriebssystemen sogar Potential zur Steigerung der Testgeschwindigkeit besteht.

Gewindespindel-Mutter-Antriebe

Bei langsamem, präzisen Verfahrbewegungen bei gleichzeitig größerer mechanischen Belastung finden sich in 3D-Druckern Gewindespindel-Mutter-Antriebe. Diese werden dort oftmals für vertikale Achsen, die das Eigengewicht des Druckkopfes und eventuell weiterer Linearachsen tragen müssen, eingesetzt [54] (vgl. Abbildung 3.2b).Auch bei hochpräzisen Werkzeugmaschinen, bei denen hohe Vorschubkräfte benötigt werden, findet sich diese Antriebstechnik in verschiedenen Varianten wieder [50, S. 82-94]. Bei diesem Antriebsprinzip wird eine rotatorische Bewegung durch eine schraubenartige Konstruktion in eine Linearbewegung umgewandelt. Auch hierbei kommen als rotatorische Antriebe in 3D-Druckern meist Schrittmotoren zum Einsatz. Wie auch bei den Linearführungen gibt es Varianten mit Wälzkörpern zwischen Spindel und Mutter, wie etwa die Kugelgewindespindel sowie Konstruktionen, bei denen diese Bauteile direkt aneinander gleiten. Eine bei 3D-Druckern aufgrund der sehr geringen Kosten weit verbreitete Ausprägung der zuletzt genannten Variante ist der Trapezgewindespindel-Antrieb. Als Gleitpaarung wird dabei eine Stahlspindel mit einer Mutter aus Bronze oder Kunststoff verwendet. Im Weiteren

⁹https://reprap.org/wiki/RepRap_Options#Belts_and_Pulleys

wird lediglich der Trapezgewindespindel-Antrieb betrachtet.

Ein großer Vorteil dieser Antriebstechnik ist die sehr hohe Steifigkeit, gepaart mit einer hohen Genauigkeit. Die Positioniergenauigkeit wird durch die Steigungsgenauigkeit der Gewindespindel, das Spiel, d.h. die Toleranzen zwischen Mutter und Gewindespindel und die Winkelauflösung des Antriebsmotors bestimmt. Durch Einsatz einer vorgespannten Mutter kann das Spiel im Antriebsstrang weitestgehend entfernt werden. Der durch das Fertigungsverfahren bestimmte Fehler in der Steigung der Spindel ist statisch und kann durch eine initiale Kalibrierung ebenfalls kompensiert werden. Durch eine direkte Ankopplung des Antriebsmotors an die Gewindespindel entsteht ein sehr kompaktes Antriebssystem, das im Gegensatz zum Riementrieb keine mechanische Belastung auf die umliegenden Strukturen ausübt. Trapezgewindespindeln sind wie Riemenantriebe verschleiß- und wartungsarm.

Als Nachteile dieser Konstruktion sind die hohen Kosten zu nennen. Diese können, je nach Fertigungsverfahren der Spindel und Mutter, ein Vielfaches eines vergleichbaren Riemenantriebs betragen. Weiterhin sind mit diesem Antriebssystem aufgrund kleiner mechanischer Übersetzungen nur geringe Verfahrgeschwindigkeiten möglich. Abhilfe können spezielle Ausführungen mit Steilgewinde oder ein vorgeschaltetes Übersetzungsgetriebe schaffen.

Im Hinblick auf die Auslegungskriterien, Positioniergenauigkeit und maximale Verfahrgeschwindigkeit, ist für einen Trapezgewindespindel-Antrieb die Steigung $h \in \mathbb{R}^+$ die wichtigste Kenngröße. Sie beschreibt, welchen Weg die Mutter bei einer vollständigen Umdrehung der Gewindespindel zurücklegt. Mit dieser Größe lässt sich der Zusammenhang zwischen dem linearen Verfahrweg l des Schlittens und dem Motorwinkel α_{Mot} in Grad wie folgt formulieren

$$l = h \cdot \frac{\alpha_{Mot}}{2\pi}$$

Die theoretische Positioniergenauigkeit l_{min} bestimmt sich damit zu

$$l_{min} = \frac{h}{s \cdot s_{Mikro}}$$

Wie bereits für den Riemenantrieb gezeigt, lässt sich auch die maximale Verfahrgeschwindigkeit v_{max} in Abhängigkeit der Spindelsteigung h und der maximalen Motordrehzahl $v_{Mot,max}$ darstellen:

$$v_{max} = \frac{h \cdot v_{Mot,max}}{60}$$

Auch für einen Antrieb mittels eines Trapezgewindespindel-Mutter-Antriebs wurde eine überschlägige Auslegung hinsichtlich der geforderten Positioniergenauigkeit und maximaler Verfahrgeschwindigkeit durchgeführt. Dabei wurde ebenfalls der im vorigen Kapitel ausgewählte NEMA 14 Schrittmotor zu Grunde gelegt. Es wurden zwei Gewindespindeln mit jeweils 10 mm und 12 mm Außendurchmesser ausgewählt. Bei diesen Größen ist noch eine kompakte Konstruktion möglich. Für genormte, recht kostengünstig erhältliche Trapezgewindespindel sind die Werte der Spindelsteigungen definiert und liegen für die ausgewählten Durchmesser bei

Motor	h in mm	s	s_{Mikro}	$v_{Mot,max}$ in min^{-1}	l_{min} in mm	v_{max} in mm/s
NEMA 14	2	200	1	400	0,01	13,3
NEMA 14	3	200	2	400	0,0075	20,0
NEMA 14	2	200	16	400	$6 \cdot 10^{-4}$	13,3
NEMA 14	3	200	16	400	$9 \cdot 10^{-4}$	20,0

Tabelle 3.2.: Auslegung des Trapezgewindespindel-Mutter-Antriebs

$p = 2 \text{ mm}$ bzw. $p = 3 \text{ mm}$.

Die Ergebnisse der Auslegung finden sich in Tabelle 3.2. Wie aus den ersten beiden Zeilen ersichtlich ist, kann mit diesem Antriebssystem eine sehr hohe Auflösung bei der Positionierung erreicht werden. Die geforderte, theoretische Positioniergenauigkeit wird bei einer Spindelsteigung von 2 mm bereits im Vollschrittbetrieb des Motors erreicht. Sie kann durch ein entsprechend hoch gewähltes Mikrostepping weiter verfeinert werden (vgl. Zeile 3 und 4). Allerdings sind die erreichten maximalen Verfahrgeschwindigkeiten aufgrund der geringen Steigung ebenfalls mit 13,3 bzw. 20,0 mm/s recht niedrig. Für die Erreichung der geforderten Verfahrgeschwindigkeit von 100 mm/s müssten daher deutlich teurere Steigengewindespindeln oder alternativ Motoren mit höherer Maximaldrehzahl eingesetzt werden.

3.4.4. Softwareseitige Ansteuerung

Wie in Kapitel 3.4.3 beschrieben, sind für den Antrieb der Linearachsen Schrittmotoren vorgesehen. Durch elektrische Impulse werden diese in diskreten Schritten rotiert. Die Anzahl der Schritte pro Umdrehung ist hierbei durch den Aufbau des Motors vorgegeben. Durch entsprechende Wahl der Steuerimpulse können jedoch die Einzelschritte weiter aufgelöst werden. Je nach Anzahl der sogenannten Mikroschritte wird zwischen den Betriebsmodi Voll-, Halb- und Mikroschrittbetrieb unterschieden [50, S. 25-29].

Zur Erzeugung dieser Impulsfolgen kommen sogenannte Schrittmotortreiber zum Einsatz. Es handelt sich dabei um eigenständige Einheiten. Diese werden häufig über zwei Steuerleitungen in Form einer Schritt-Richtung-Schnittstelle angesteuert. Auf einer Leitung werden Impulse zum Takten des Schrittmotors angelegt. Durch den Pegel der zweiten Leitung wird die Drehrichtung definiert. Neuere Modelle verfügen über Schnittstellen für Bussysteme, wie etwa CAN oder SPI.

Bei einem kartesischen Roboter sind zur Führung eines Aktors entlang einer vorgegebenen Kurve im Raum mehrere Achsen gleichzeitig zu verfahren. Eine präzise, aufeinander abgestimmte Ansteuerung der entsprechenden Antriebssysteme ist daher notwendig. Im Bereich des 3D-Drucks existieren hierfür etliche OpenHardware Projekte für Kinematiken mit bis zu 5 Achsen.

Bekannte Projekte sind etwa das RAMBo¹⁰ oder das Smoothieboard¹¹. Diese Steuerplatinen verfügen entweder über bereits integrierte Schrittmotortreiber oder bieten Steckplätze für diese an. Die entsprechende Firmware wird bei manchen Projekten, wie etwa dem Smoothieboard, ebenfalls bereitgestellt. Es existieren aber auch eigenständige OpenSource Projekte hierfür. Ein Beispiel ist die für 3D-Drucker entwickelte Firmware Marlin¹². Aufgabe der Firmware ist zum einen die Bahnerzeugung durch entsprechende Ansteuerung der Antriebssysteme, aber auch die für 3D-Drucker notwendige Temperaturregelung des Heizbettes und Extruders.

Fahr- und Steuerbefehle werden durch genormte, meist aber zusätzlich um spezielle Befehle erweiterte, textuelle Kommandos formuliert. Die oftmals auch als *G-Codes* bezeichnete Programmiersprache ist etwa in Deutschland durch DIN 66025 genormt [56, S. 207-218]. Die Steuerplatinen sind in der Lage, eine Reihe von Steuerbefehlen aus einer Datei abzuarbeiten oder diese einzeln über Schnittstellen, etwa eine serielle UART-Schnittstelle, zu empfangen.

Für den umgesetzten Prototypen wurde eine abgewandelte Version des Smoothieboards genutzt. Das *BIGTREETECH SKR V1.3 Board*¹³ verfügt über einen 32 Bit LPC1768 Arm® Cortex-M3 Prozessor mit 100 MHz Taktfrequenz. Schrittmotortreiber können über SPI, UART oder die beschriebene Zweidraht Schritt-Richtung-Schnittstelle angesteuert werden. Als Schrittmotortreiber wurden Controller des Typs TMC2130 der Firma TRINAMIC¹⁴ eingesetzt. Diese können Motoren mit einer Eingangsspannung von 5 bis 46 V und einem Strom von bis zu 2,5 A ansteuern. Sie erlauben ein Mikrostepping mit bis zu 256 Mikrosschritten pro Schritt und enthalten zusätzliche Features, wie etwa eine Hinderniserkennung und dynamische Lasterkennung.

Die OpenSource Firmware Marlin¹⁵ erlaubt die Ansteuerung der gewählten Schrittmotortreiber. Durch einen modularen Aufbau ist sie außerdem sehr leicht zu erweitern. Sie wurde daher auf der gewählten Steuerplatine als Firmware eingesetzt.

Die Architektur zur Ansteuerung der Kontaktierhardware mit allen eingesetzten Komponenten und die verwendeten Protokolle zeigt Abbildung 3.6. Da jede Steuerplatine nur bis zu 5 Achsen ansteuern und auch die eingesetzte Firmware nur für die Bahnsteuerung eines Aktors ausgelegt ist, sieht der Gesamtaufbau für jede fahrende Einheit eine separate Steuerplatine vor. Die einzelnen Platten werden von einem Host-Rechner per UART über USB angesprochen. Funktionen zur Steuerung des Gesamtsystems werden über eine Application Programming Interface (API) bereitgestellt. Die API wurde als eigenständiges Modul in der Programmiersprache Python umgesetzt. Sie abstrahiert sowohl die Kommunikation mit den einzelnen Steuerplatten, als auch den Zugriff auf das eingesetzte Kamerasytem. Weiterhin sind etwa Funktionalitäten zur automatischen Identifizierung und Konfiguration der einzelnen Einheiten integriert.

¹⁰<https://github.com/ultimachine/RAMBo>

¹¹<https://github.com/Smoothieware/Smoothieboard>

¹²<https://github.com/MarlinFirmware/Marlin>

¹³<https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3>

¹⁴<https://www.trinamic.com/products/integrated-circuits/details/tmc2130/>

¹⁵<https://github.com/MarlinFirmware/Marlin>

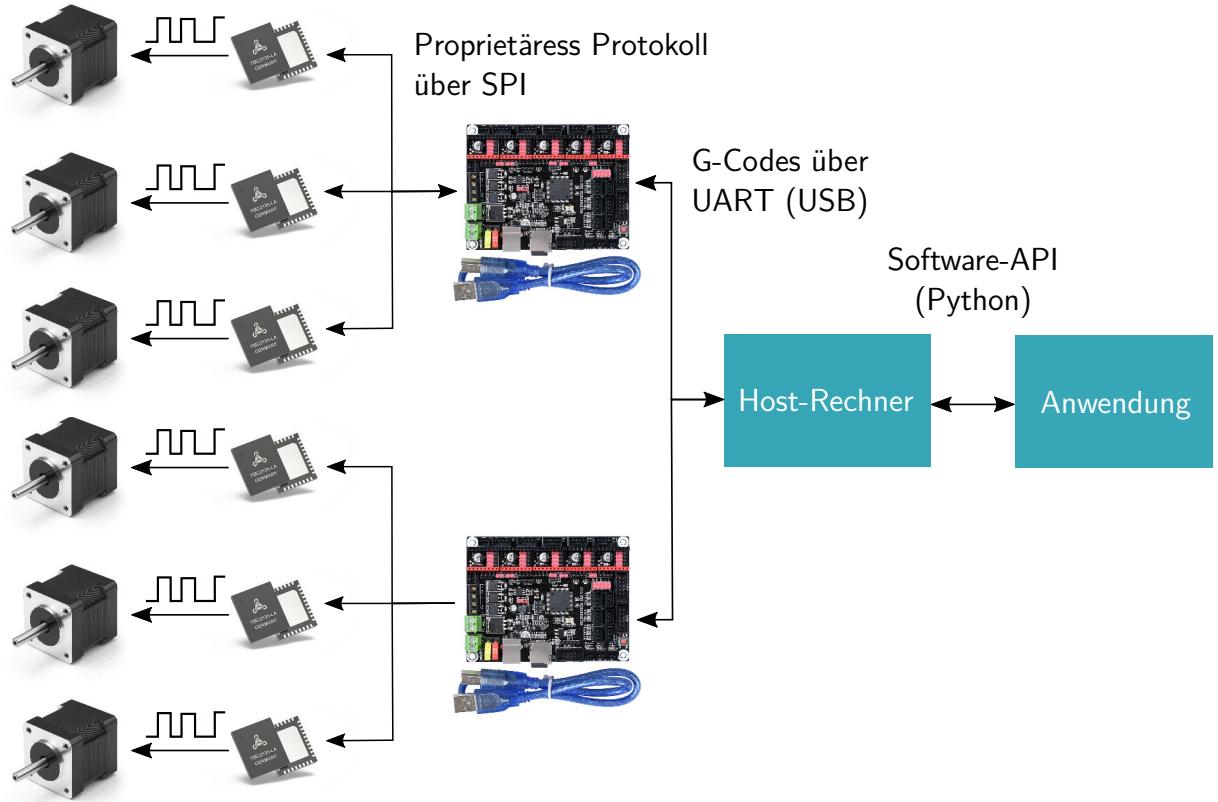


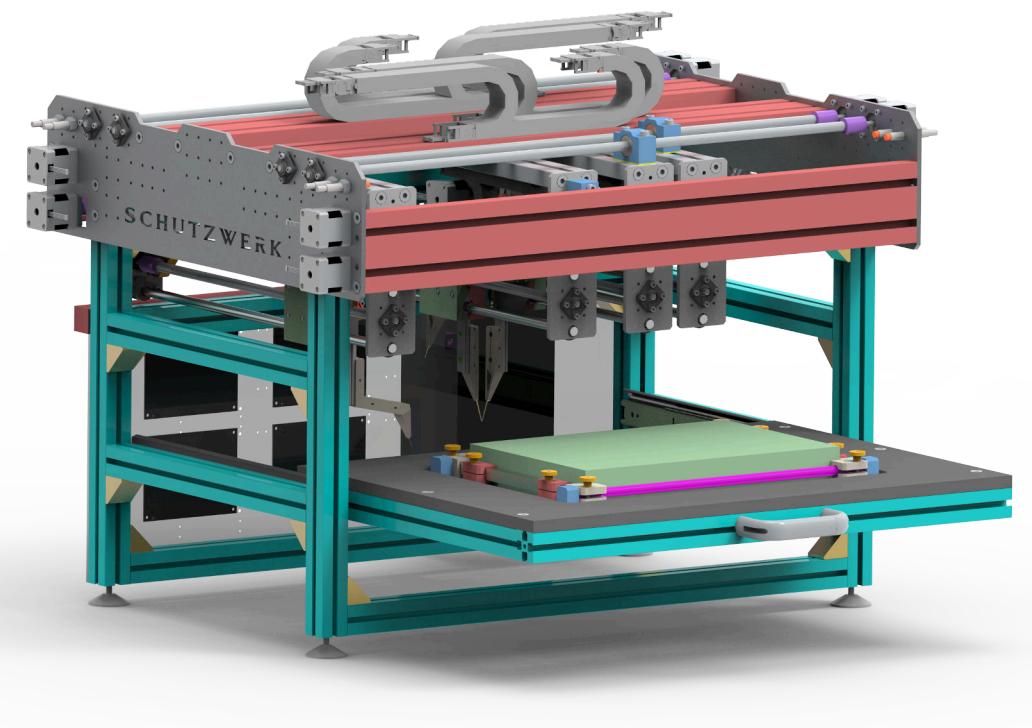
Abbildung 3.6.: Architektur zur Ansteuerung der Kontaktierhardware (Bildquellen: Schrittmotor [57], Treiber [58], Platine [59])

3.4.5. Finaler Prototyp

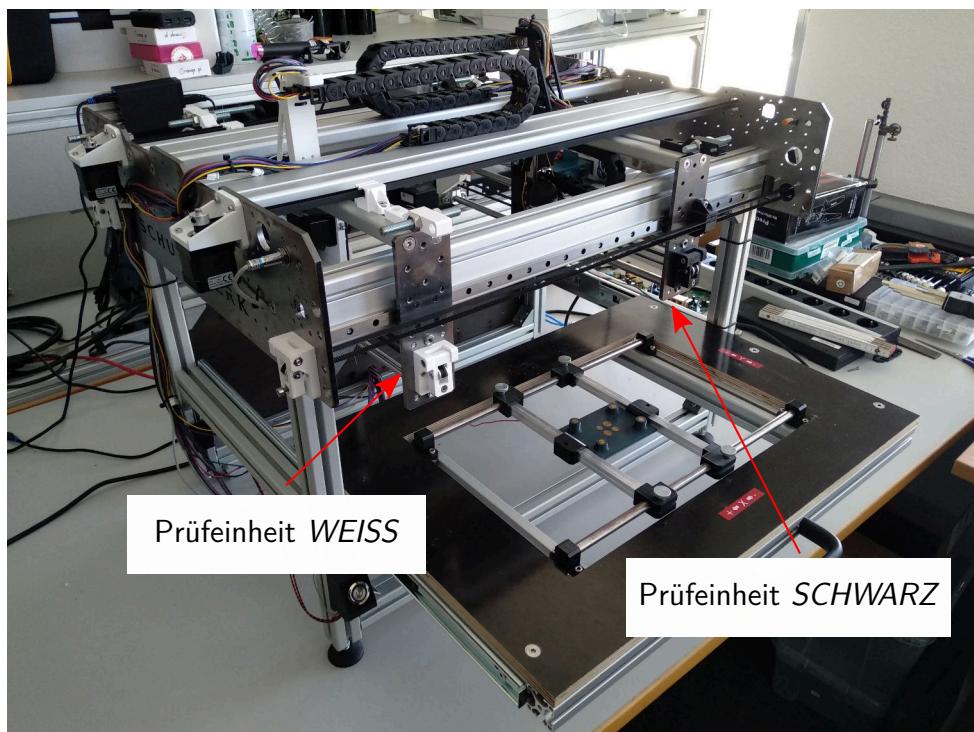
Anhand der in den vorigen Kapiteln beschriebenen Vorüberlegungen, wurde in einem iterativen Prozess der finale Aufbau der Kontaktierhardware entworfen. Dazu wurde die für Studenten kostenlose CAD-Software Solid Edge der Firma Siemens verwendet¹⁶, da diese einen gegenüber existierender OpenSource-Software deutlich erweiterten Funktionsumfang besitzt. Zwischenstände und verschiedene untersuchte Konstruktionsvarianten sind in Anhang D abgebildet. Das endgültige Design zeigt Abbildung 3.7a. Entsprechend diesem Entwurf wurde ein realer Prototyp mit zwei fahrenden Einheiten aufgebaut. Dieser ist in Abbildung 3.7b zu sehen. Er besteht aus dem Grundrahmen, sowie zwei fahrbaren Einheiten zur Platzierung von jeweils einer Prüfspitze. Angelehnt an die Farbgebung der 3D-gedruckten Bauteile werden die einzelnen Kontaktiereinheiten als Prüfeinheit *SCHWARZ* und Prüfeinheit *WEISS* bezeichnet (siehe Abbildung 3.7b). Das fahrbare Kamerasytem wurde an der Prüfeinheit *SCHWARZ* montiert.

Die Basis des Prototyps bildet ein Rahmen aus Aluminiumprofilen. Eine ausziehbare Schublade, auf welcher die Aufnahme für den zu untersuchenden PCB montiert ist, ermöglicht eine sehr einfache Be- und Entladung. Eventuell benötigte Peripherie zum Betrieb des Untersuchungsobjekts kann durch die Öffnung in der Schublade nach unten oder zu den Seiten hin

¹⁶<https://solidedge.siemens.com/de/>



(a) Finaler CAD-Entwurf



(b) Aufgebauter Prototyp

Abbildung 3.7.: Prototyp der Kontaktierhardware

angeschlossen werden. Die gesamte Steuerelektronik sowie die Stromversorgung ist auf einer Platte an der Rückseite des Aufbaus montiert.

Alle fahrenden Einheiten sind auf einem in sich geschlossenen Rahmen, bestehend aus jeweils zwei gelaserten Stirnplatten und Rechteckaluminiumprofilen, montiert. Sie teilen sich eine gemeinsame, seitlich an den Aluminiumprofilen angebrachte Profilschienenführung der Größe MR-12. Die Einheiten sind hintereinander angeordnet, was eine Einschränkung der Beweglichkeit in den Einheiten mit sich bringt: Die hinteren beiden Verfahreinheiten können die vorderen nie überholen. Durch geschickte Konstruktion der Aufnahmen für die Prüfspitzen ist es aber trotzdem möglich, alle Spitzen in einem Punkt bzw. einer Linie in X-Richtung anzurichten (vgl. Abbildung 3.9). Durch einen Multiplexer am Ausgang der Signalleitungen der Prüfspitzen und geschicktes Durchschalten, können aber auch bei dieser Achsanordnung theoretisch vier beliebige Punkte auf dem zu testenden PCB platziert werden.

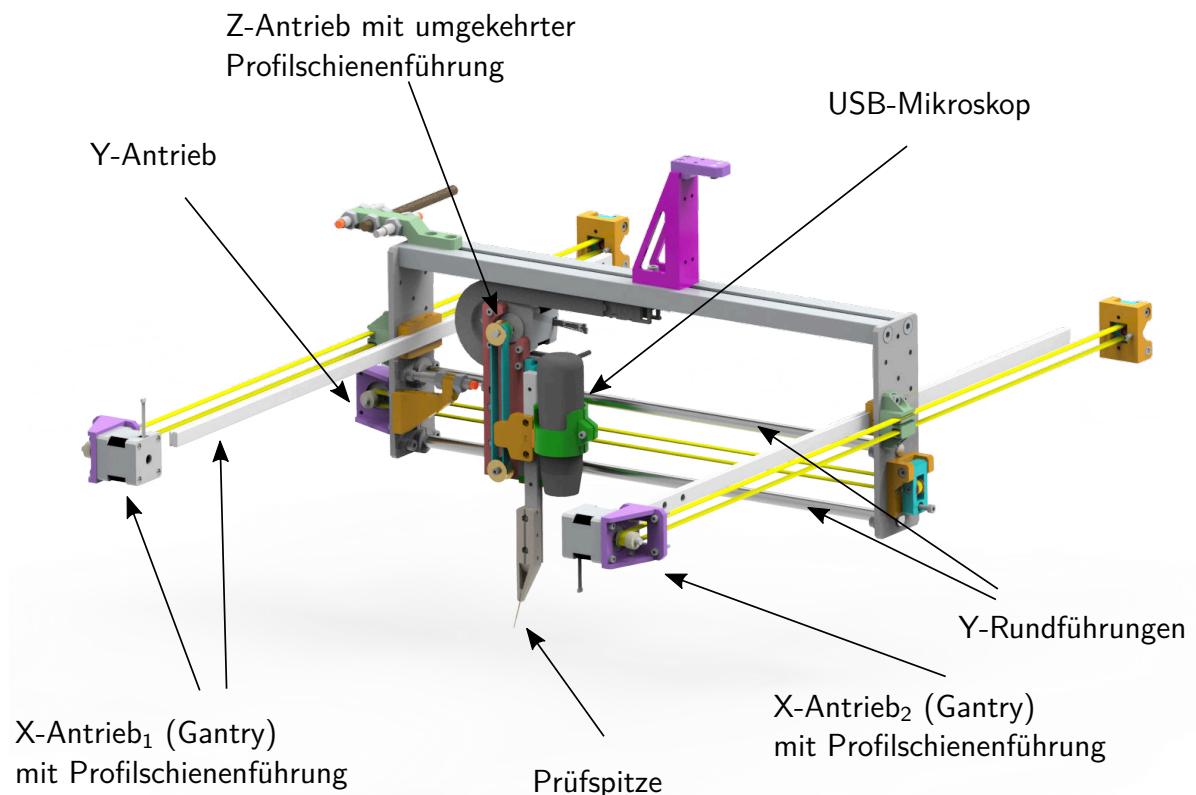


Abbildung 3.8.: Fahrbare Einheit mit Riemenantrieben in allen Achsen

Um eine möglichst platzsparende Konstruktion zu erzielen, wurden die fahrenden Einheiten kassettenförmig, schmal und hoch konstruiert. Dies erlaubt es, auf relativ kleinem Raum möglichst viele dieser Einheiten unterzubringen. Den Aufbau einer Achse zeigt Abbildung 3.8. Die schmale Konstruktion in Richtung der X-Achse macht einen beidseitigen Antrieb in dieser Achse notwendig, um ein Verkippen und damit ein Verklemmen der Führungen zu verhindern.

Ein solcher Doppelantrieb wird auch als Gantrybauweise bezeichnet [50, S. 10]. Beide Antriebe müssen hierbei synchron bewegt werden. Die Aufgabe der Antriebssynchronisation wird im aufgebauten Prototyp softwaretechnisch durch die Firmware der Steuerplatine realisiert.

Der fahrbare X-Schlitten besteht aus zwei gelaserten Seitenplatten. Diese werden an der Oberseite über einen in Form eines Aluminiumprofils umgesetzten Balken miteinander verbunden. An der Unterseite erfolgt eine weitere Verbindung der beiden Seitenplatten über die zwei geklemmten Führungsstangen der Y-Rundführungen. Durch den Einsatz dieses Führungssystems konnte auf eine aufwendige Schlittenkonstruktion verzichtet werden. Da das von dieser Achse zu tragende Gewicht nur sehr gering ist, genügen an dieser Stelle Führungsstangen mit einem Durchmesser von 10 mm.

Der quer fahrende Y-Schlitten ist als 3D gedrucktes Teil ausgeführt. Er besteht aus zwei Hälften, was eine einfache Montage der aus tribologisch optimiertem, reibungssarmen Kunststoff gefertigten Führungsbuchsen der Y-Rundführung erlaubt. An dem Y-Schlitten ist neben der vertikal fahrenden Z-Achse auch das Kamerasystem befestigt. Dieses besteht aus dem bereits in den vorigen Kapiteln erwähnten zylinderförmigen USB-Mikroskop.

Die Z-Achse wird durch einen sehr kompakten Riemenantrieb bewegt. Zur Führung kommt ein Profilschienenführungssystem der Größe MR-12 zum Einsatz. Durch eine umgekehrte Anordnung, bei welcher der Schlitten der Z-Führung fest auf dem Y-Schlitten montiert ist und dafür die Führungsschiene bewegt wird, ist diese Einheit ebenfalls sehr kompakt, und es können nahezu alle beweglichen Bauteile aus dem Arbeitsraum gefahren werden.

Die Führungsschiene dient außerdem zur Aufnahme der 3D gedruckten Prüfspitzenhalterung. Durch Nutzung der Montagelöcher in der Führungsschiene ist ein Wechsel dieser Aufnahme sehr einfach und mit wenigen Handgriffen durchführbar. Damit die vier Prüfspitzen in der größten Ausbaustufe möglichst dicht nebeneinander platziert werden können, wie es beispielsweise bei TQFP Packages der Fall ist, wurden die Spitzen in einem Winkel von 10 bzw. 25° um die Y-Achse gedreht. Dies zeigt Abbildung 3.9, in welcher alle vier Prüfspitzen auf einen Punkt gefahren dargestellt sind.

Die Konstruktion wurde so modular gehalten, dass in der X- und Y-Achse entweder der in Kapitel 3.4.3 vorgestellte Riemenantrieb oder Trapezgewindespindel-Mutter-Antrieb eingesetzt werden kann. In Abbildung 3.8 ist eine Einheit mit Riemenantrieb in beiden Achsen zu sehen. Abbildung 3.7a zeigt bei ansonsten identischer Konstruktion eine Ausführung mit Trapezgewindespindel-Mutter-Antrieb. Der real aufgebaute Prototyp enthält derzeit Riemenantriebe und entspricht daher der in Abbildung 3.8 dargestellten Version.

Dadurch, dass derzeit nur zwei fahrbare Einheiten in dem umgesetzten Prototypen eingebaut sind, ergibt sich ein erweiterter Arbeitsraum für diese Einheiten von $420 \times 200 \times 50$ mm in der X-, Y- und Z-Achse.

Das eingesetzte, fahrende Kamerasystem des Prototyps besteht aus einem USB-Mikroskop, das auf einem der Y-Schlitten montiert ist. Es ist in einer Höhe von etwa 100 mm über der PCB-Aufnahme angebracht. In Kombination mit der eingebauten Optik ergibt sich dadurch ein

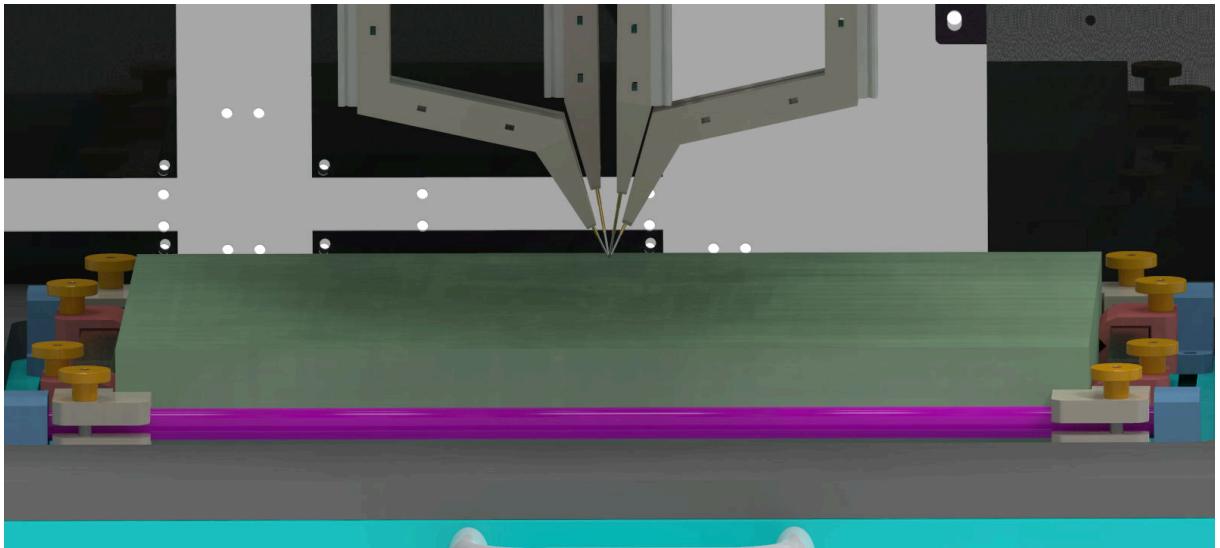


Abbildung 3.9.: Anordnung der Prüfspitzen

FOV von ca. $30(X) \times 40\text{ mm}(Y)$. Die geforderte minimale Pixelauflösung von $0,085\text{ mm}/\text{Pixel}$ wird dadurch schon bei einer Auflösung von 640×480 Pixel mit $0,0625\text{ mm}/\text{Pixel}$ erreicht. Eine Livebildübertragung mit 30 FPS ist daher auch über eine USB2.0-Verbindung noch möglich.

Die Montage des Prototyps ist relativ einfach, und konstruktionsbedingt wird hierfür nicht viel Werkzeug benötigt. Alle Bauteile werden dabei grob ausgerichtet montiert. Eine aufwendige Einrichtung mit teuren Messmitteln ist nicht notwendig. Den Abgleich der Achsen untereinander sowie den Ausgleich von statischen Fertigungs- und Montagetoleranzen übernimmt eine neu entwickelte Kalibrierungsroutine, welche in Kapitel 4 vorgestellt wird. Diese Vorgehensweise wurde gewählt, um den Nachbau des Systems im Sinne des OpenHardware-Gedankens zu vereinfachen und ebenfalls die Kosten möglichst gering zu halten.

Der mit ca. 1000 € festgelegte Kostenrahmen konnte nicht ganz eingehalten werden. Die Kosten für den Grundrahmen belaufen sich auf ca. 500 €. Jede Prüfeinheit, bestehend aus einem 3-Achsenystem ohne Kosten für die 3D gedruckten Teile, kostet zusätzlich ca. 650 €. Die Gesamtkosten für den aufgebauten Prototypen belaufen sich damit auf etwa 2300 € inkl. Mehrwertsteuer. Es sind keine Arbeitskosten für Montage und manuelle Bearbeitung einiger Teile enthalten. Da der Prototyp im Rahmen einer studentischen Arbeit entstanden ist, wurden von der Firma igus¹⁷ einige Teile, wie etwa die eingesetzten Energieketten und Kunststoffgleitlagerbuchsen, als kostenlose Muster zur Verfügung gestellt.

3.4.6. Beobachtungen

In diesem Kapitel werden einige Erkenntnisse, die bei der Montage und ersten Tests des Prototyps gewonnen wurden, zusammengefasst.

Neben dem final eingesetzten Riemenantrieb in der X- und Y-Achse wurden auch Tests

¹⁷<https://www.igus.de/>

mit einem Trapezgewindespindel-Mutter-System durchgeführt. Dabei kamen vorgespannte Gewindemuttern aus Kunststoff der Firma igus¹⁸ zum Einsatz. Als Gantryausführung in der X-Achse aber auch in einer einfachen Ausführung in der Y-Achse ist dieses Antriebssystem extrem steif. Durch die Vorspannung in den Muttern ergibt sich ein nahezu spielfreies Antriebssystem, dessen Positioniergenauigkeit lediglich durch den Steigungsfehler der Gewindespindel und den Winkelfehler des Antriebsmotors bestimmt ist.

Die Vorspannung in den Gewindemuttern erhöht die Reibung und somit auch das benötigte Motordrehmoment. Die überschlägig dimensionierten Motoren reichten für dieses Antriebssystem bei höheren Geschwindigkeiten nicht mehr aus. Im Gantryantrieb der X-Achse kam es in den Beschleunigungsphasen durch das mit zunehmender Drehzahl abfallende Motordrehmoment¹⁹, öfters zum Blockieren eines Motors. Dies führte zu einer Kettenreaktion, wodurch der zweite Antrieb kurz darauf blockierte. Um dieses Antriebssystem daher produktiv einsetzen zu können, müssen deutlich größere Motoren mit höherem Drehmoment bei gleichzeitig höherer Drehzahl verwendet werden. Auch die exakte Ausrichtung beider Antriebe zueinander erwies sich als kritischer Faktor. Ist der Gantryantrieb nicht von Anfang an mechanisch spannungsfrei montiert, kommt es vermehrt zu Blockaden und Verklemmungen.

Ebenfalls negativ fiel beim Einsatz des Trapezgewindetriebs eine starke, von der Motordrehzahl abhängige Geräuschentwicklung in der X- und Y-Achse auf. An dem gesamten Gestell und auch den fahrenden Schlitten konnten deutlich Vibrationen wahrgenommen werden. Ursache hierfür könnte die direkte Montage der Antriebsmotoren auf die großen, flächigen Seitenplatten sein. Schwingungen aus den Motoren bzw. des Antriebsstrangs übertragen sich dadurch direkt auf das gesamte Gestell. Die großen Seitenplatten wirken wie große Membranen eines Lautsprechers und verstärken die Geräuschentwicklung. Dieses Problem könnte durch die nachträglich eingebauten Verstrebungen an der Oberseite behoben worden sein (vgl. Abbildung 3.7a, rötliche Balken). Mit der überarbeiteten Konstruktion wurden jedoch aus zeitlichen Gründen keine Tests mehr mit diesem Antriebssystem durchgeführt.

Bei schnellen Verfahrbewegungen, sowohl unter Einsatz des Riemen- als auch des Trapezgewindetriebs konnten Schrittverluste bemerkt werden. Dies zeigte sich dadurch, dass die Schlitten in ihren Endpositionen auf die eingebauten Endanschläge auffuhren. Es ist zu vermuten, dass die eingesetzten Schrittmotoren unterdimensioniert sind. Da keine Anforderungen hinsichtlich der Dynamik gestellt wurden und dementsprechend keine Auslegung bezüglich des Motormoments erfolgte, ist dies durchaus möglich. Zukünftig sollten daher auch Kriterien für das dynamische Verhalten festgelegt und die Motorauslegung überarbeitet werden.

Neben der Profilschienenführung wurde auch eine Gleitführung des Herstellers igus²⁰ erprobt. Durch identische Einbaumaße kann das Führungssystem problemlos ausgetauscht werden. Das System ist, im Gegensatz zu Profilschienenführungen, nahezu wartungsfrei. Es wurde, gemäß den Herstellerangaben eine Fest-Loslager-Ausführung mit voreingestelltem Spiel getestet. Mit

¹⁸<https://www.igus.de/product/847>

¹⁹vgl. Anhang C

²⁰<https://www.igus.de/product/930>

dieser Gleitführung konnte eine deutliche Beruhigung der gesamten Mechanik hinsichtlich der auftretenden Schwingungen und Geräuschentwicklung erzielt werden. Dies ist durch die dämpfende Wirkung der Gleitelemente aus Kunststoff zu erklären. Das voreingestellte Lagerspiel war allerdings zu groß, was zum regelmäßigen Verklemmen des Gantryantriebs der X-Achse führte. Im finalen Prototyp wurden daher wieder Profilschienenführungen eingesetzt. Das Gleitführungssystem könnte aber vermutlich bei richtig eingestelltem Lagerspiel ebenso eingesetzt werden.

Die Z-Achse ist insgesamt recht langsam. Um innerhalb der kurzen Hubstrecke von nur 50 mm dynamisch zu verfahren, sind hohe Beschleunigungskräfte und damit ein großes Motormoment notwendig. Durch Einsatz eines größer dimensionierten Antriebsmotors könnte vermutlich eine deutliche Dynamiksteigerung erreicht werden.

Das eingesetzte Kamerasystem erwies sich als nicht optimal. Über den zur Verfügung gestellten Treiber sind nur sehr wenige Parameter der Kamera einstellbar. Ein Zugriff auf wichtige, die Qualität der Aufnahmen stark beeinflussende Parameter, wie etwa den automatischen Weißabgleich oder die Belichtungszeit, war leider nicht möglich. Aufnahmen mit scharfen Kanten, wie sie etwa bei der Kamerakalibrierung²¹ benötigt werden, waren nur schwer zu erreichen. Wurde die Umgebungsbeleuchtung nicht optimal eingestellt, kam es öfters zu starkem Bildrauschen. Der von der Kamera intern eingesetzte Komprimierungscodec führte außerdem bei höheren Bildauflösungen zu starken Aliasingeffekten mit entsprechender Treppenbildung. Auch dies erschwerte die Kamerakalibrierung zusätzlich. Eine Umstellung auf einen anderen, weniger stark komprimierenden Codec könnte dieses Problem beheben. Andererseits nimmt mit zunehmender Bildauflösung auch die Menge der zu übertragenden Daten zu, was sich negativ auf die Bildübertragungsrate des Live-Streamings auswirkt. Aus den genannten Gründen wurde ein neues Kamerasystem für den Prototypen bestellt, welches in ersten Tests deutlich bessere Aufnahmen lieferte. Leider war es nicht mehr möglich, im Rahmen dieser Arbeit diese neue Kamera in den Prototypen zu integrieren.

3.5. Zusammenfassung und Ausblick

Resultierend aus den im vorigen Kapitel beschriebenen Beobachtung werden an dieser Stelle Vorschläge für mögliche Verbesserungen und Weiterentwicklungen des entwickelten Prototyps gegeben.

Weiterentwicklung der Mechanik

Die Antriebsauslegung sollte nach Definition der Anforderungen hinsichtlich der Dynamik der Kontaktierhardware neu durchgeführt werden. Dadurch ließen sich sicher einige der aufgetretenen Probleme, wie etwa Schrittverluste und damit Ungenauigkeiten in den Achsen von vornherein beheben.

²¹siehe Kapitel 4.2.1

Für Tests von komplexeren Bussystemen, wie etwa SPI, müssen weitere Prüfspitzen und damit auch fahrbare Einheiten ergänzt werden. Durch die modulare Konstruktion ist dies sehr einfach möglich. Die benötigten Konstruktionsdaten sind bereits vorhanden. Auch eine beidseitige PCB-Kontaktierung ist mit dem entworfenen System denkbar. Da alle fahrenden Einheiten auf einem geschlossenen Rahmen angebracht sind, könnte dieser einfach dupliziert und um 180° gedreht montiert werden. Damit stünden insgesamt acht frei platzierbare Prüfspitzen für Sicherheitstests zur Verfügung.

Die Prüfspitzen sind derzeit leicht gegenüber der Y-Achse gedreht montiert. Eine Kontaktierung aller Komponentenpins eines QFP Chips mit einer einzigen Spalte ist damit vermutlich nicht möglich, da es durch die schräge Anordnung auf manchen Seiten zu Kollisionen der Prüfspitze mit dem Chip *Package* kommen kann. Abhilfe hierbei könnte eine rotierbare Prüfspitze schaffen. Der Winkel der Prüfspitze bleibt gleich. Durch eine zusätzliche Rotationsachse um die Z-Achse des Achssystems könnte ein optimaler Kontaktierwinkel an allen Seiten eines QFP oder auch eines schräg platzierten Chips gewährleistet werden. Die Aufnahme der Prüfspitze ist durch die gewählte Konstruktion sehr leicht austauschbar. Der Konstruktions- und Steuerungsaufwand für eine solche Achse ist vergleichsweise gering. Jedoch wird die initiale Kalibrierung einer solchen Einheit komplexer und ist mit der entwickelten Kalibrierungsroutine derzeit nicht möglich.

Weiterentwicklung der Steuerelektronik

Zur Vermeidung von Kollisionen der fahrbaren Einheiten untereinander sollten sowohl softwareseitig, als auch durch entsprechende mechanische Maßnahmen, etwa in Form von Endanschlägen und Auffahrpuffern, Schutzmechanismen ergänzt werden. Dieser Punkt ist umso wichtiger, je mehr fahrende Einheiten in das System eingebaut werden.

In diesem Zusammenhang wäre auch eine Überarbeitung der elektronischen Ansteuerung der Achsen denkbar. So könnten etwa die Steuerplatten in einer Master-Slave Topologie angeordnet werden. Eine Kommunikation mit dem gesamten Achssystem würde dann über eine Master-Platine ablaufen. Diese würde die Fahrbefehle decodieren und die Weiterleitung an die restlichen Platten übernehmen. Die Software des Host-Rechners und Applikationen auf höherer Abstraktionsebene würden dadurch vereinfacht. Zur Umsetzung dieser Topologie müsste allerdings die Firmware der Steuerplatten grundlegend überarbeitet werden. Ein alternativer Ansatz wäre, alle Schrittmotortreiber über einen einzigen Mikroprozessor anzusteuern. Die ausgewählten Treiber werden derzeit über eine SPI-Schnittstelle angesprochen. Durch Durchschleifen der Datenleitungen wird lediglich pro Motortreiber eine zusätzliche Chip Select Leitung benötigt. Hierfür stehen auf den Steuerplatten noch genügend *General Purpose IO (GPIO)*-Pins zur Verfügung. Die zentrale Steuerung durch einen einzigen Mikrocontroller könnte sogar eine schrittgenaue Synchronisierung der einzelnen Achsen ermöglichen. Weiterhin würden dadurch die Anschaffungskosten für die Steuerplatten entfallen. Auch bei diesem Ansatz wären tiefgreifende Änderungen an der Firmware notwendig.

Der aufgebaute Prototyp enthält bisher lediglich Sensorik für eine Durchgangsprüfung. Ent-

sprechend können derzeit nur Analysen zur Rekonstruktion der Netzlisten durchgeführt werden. Um Sicherheitsanalysen durchführen zu können, muss daher zusätzliche Auswertelektronik zusammen mit einem Multiplexer zum wahlfreien Routen der Prüfspitzen zur Analysehardware ergänzt werden. Entsprechende Anforderungsanalysen gängiger Sicherheitstests sind hierfür durchzuführen.

Weiterentwicklung des Kamerasytems

Bezüglich der Kontaktierhardware sind einige Erweiterungen vorstellbar, welche wichtige Funktionalitäten mit sich bringen. Eine sehr sinnvolle Ergänzung wäre ein Autofokussystem für das Kamerasytem. Der Tiefenschärfebereich der eingesetzten Kamera ist relativ gering. Um optimale Bildaufnahmen zu erzeugen, könnte ein mechanischer Autofokus in das System integriert werden. Dabei ist ein Mechanismus anzustreben, bei dem das gesamte Kamerasytem und nicht nur die Linse bewegt wird. Dies hat den Vorteil, dass die im nächsten Kapitel vorgestellte Kamerakalibrierung nur einmalig durchgeführt werden muss.

Wie bereits in Kapitel 3.4.6 beschrieben wurde, ist das ausgewählte Kamerasytem nicht optimal. Der Einbau eines besseren, hochwertigeren Systems ist daher ratsam. Eine höhere Auflösung in mm/Pixel würde es erlauben, auch noch sehr feine Strukturen zu analysieren. Durch ein System, bei welchem Parameter wie Weißabgleich oder Belichtungszeit manuell gesetzt werden können, könnte die Bildqualität insgesamt verbessert werden.

Zur Dokumentation der durchgeführten Sicherheitsanalysen wären weitere Kamerasyteme innerhalb des Arbeitsraums sinnvoll. Die kontaktierten Pins ließen sich beispielsweise durch kleine Kameras, die in der Nähe der Prüfspitzen montiert werden, sehr gut dokumentieren. Durch diese zusätzlichen Kameras wäre auch eine optisch geführte Positionierung der Prüfspitzen denkbar. Dies würde das Vertrauen in das Gesamtsystem steigern, da zu jedem Zeitpunkt die exakte Position jeder Prüfspitze sichtbar wäre.

Eine bessere Kontrolle der Lichtverhältnisse könnte die Qualität der Kameraaufnahmen verbessern. Dazu könnte der Prototyp mit einem lichtundurchlässigen Gehäuse versehen werden. Durch gezieltes Anbringen von Lichtquellen, etwa in Form von ansteuerbaren LED-Streifen, könnten dann immer ideale Beleuchtungsverhältnisse für den jeweiligen Anwendungszweck geschaffen werden.

4. Kalibrierung der Hardwareplattform

Die in Kapitel 3 vorgestellte Hardware für das automatische Kontaktieren von PCBs besteht aus mehreren autonomen Kontaktiereinheiten und einem fahrenden Kamerasytem. Mit Hilfe des integrierten Kamerasytems können hochauflöste Bildaufnahme des zu untersuchenden PCBs generiert werden. Für den Einsatz zur automatisierten Sicherheitsanalyse muss die entwickelte Hardwareplattform in der Lage sein, beliebige, in den Kameraaufnahmen selektierte Punkte automatisch anzufahren. Eine präzise Berechnung von Positionen der einzelnen Prüfspitzen aus Koordinaten des Kamerabildes ist hierzu essentiell.

In diesem Kapitel wird daher ein Verfahren beschrieben, um eine möglichst exakte Kopplung zwischen den einzelnen Kontaktiereinheiten und dem vorhandenen, fahrbaren Kamerasytem zu erreichen. Durch die Kalibrierung sollen außerdem vorhandene geometrische Fehler, bedingt durch Toleranzen in einzelnen Bauteilen oder ungenauer Montage, korrigiert werden.

Einen Überblick über aktuell eingesetzte Verfahren für diese Problemstellung gibt Kapitel 4.1. In Kapitel 4.2 werden wichtige Grundlagen der Computer Vision sowie einige für das Verständnis der nachfolgenden Kapitel notwendige mathematische Verfahren vermittelt. Anschließend wird der im Rahmen dieser Arbeit neu entwickelte Ansatz einer Kalibrationsroutine in Kapitel 4.3 vorgestellt. Auf die Details der Kalibrierung der Kontaktier- und Kameraeinheiten gehen Kapitel 4.4 und 4.5 näher ein. Die mit diesem Verfahren erzielten Ergebnisse und die dabei eingesetzte Evaluationsmethodik werden in Kapitel 4.6 diskutiert. Das Kapitel schließt mit einer praktischen Anwendung der gewonnenen Kalibrationsdaten – der Erzeugung von hochauflösten PCB-Aufnahmen mittels *Image Stitching* (Kapitel 4.7).

4.1. Stand der Technik

Die Problemstellung der exakten Positionierung eines Endeffektors im Raum kommt sehr häufig in verschiedenen Bereichen der Technik vor. Ein Beispiel hierfür sind Maschinen aus der Fertigungsindustrie, wie etwa computergesteuerte CNC-Bearbeitungszentren. Diese sind vom kinematischen Aufbau her der entwickelten Kontaktierhardware sehr ähnlich. Auch bei diesen Maschinentypen müssen oftmals mehrere Einheiten im Submillimeterbereich präzise im Raum positioniert werden, um Bauteile mit möglichst engen Fertigungstoleranzen und hohen Oberflächengüten herstellen zu können.

In seiner Arbeit stellt Weck einige Verfahren zur Kalibrierung von Werkzeugmaschinen und Beurteilung ihrer statischen und dynamischen Eigenschaften vor [60, S. 91-130]. Es werden

Methoden zur Messung und Bestimmung der geometrischen Genauigkeit von Werkzeugmaschinen aufgezeigt. Dabei kommen neben klassischen Messmitteln, wie analoge oder digitale Messuhren und Winkelmessgeräte, auch optische Verfahren, wie die präzise Längenmessung mittels Laserinterferometern, zum Einsatz. Die Verfahren setzen häufig hochpräzise gefertigte und kalibrierte Messmittel und Prüfkörper als Referenzen voraus. Die Genauigkeit einer Werkzeugmaschine wird dann relativ zu diesen Prüfkörpern mit den Messmitteln bestimmt. Als Resultat können sehr genaue Aussagen über wichtige geometrische Eigenschaften, wie etwa die Ebenheit oder Geradheit von linear bewegten Einheiten oder auch die Winkelgenauigkeit rotierender Komponenten, getroffen werden. Die dabei eingesetzten Messmittel sind jedoch meist aufgrund der hohen Komplexität in der Fertigung sehr teuer.

Auch für Kamerasysteme gibt es etablierte Kalibrationsverfahren. Hierbei wird das Kamera-Linsensystem zu dem Modell einer Lochkamera reduziert. Ziel der Kalibrierung ist die Bestimmung der intrinsischen Parameter des Kamerasystems, wie die Brennweite, den Bildhauptpunkt und vorhandene Linsenverzerrungen. Die Kenntnis dieser Parameter ist für eine korrekte Rekonstruktion von Punkten im Raum oder für eine exakte Posenbestimmung der Kamera essentiell. Aufgrund der vielfältigen Einsatzmöglichkeiten, wie die 3D-Rekonstruktionen von topographischen Karten aus Satellitenbildern [61] oder auch Augmented Reality (AR)-Anwendungen [62], ist dieses Themengebiet gut erforscht.

Zhang etwa stellt in seiner Arbeit ein Verfahren vor, bei welchem mehrere Aufnahmen eines flachen Referenzmusters in unterschiedlichen Ausrichtungen genügen, um die intrinsischen Parameter eines Kamerasystems auf einfache Art zu bestimmen [63]. Meist kommt hierbei ein gedrucktes Schachbrettmuster zum Einsatz, welches in verschiedenen Winkeln mit dem Kamerasystem aufgenommen wird. Anhand der bekannten Koordinaten der Eckpunkte der Schachbrettfelder werden anschließend die intrinsischen Kameraparameter zusammen mit den Kameraposen, die auch als extrinsische Kameraparameter bezeichnet werden, berechnet.

4.2. Grundlagen

In diesem Unterkapitel werden für den weiteren Verlauf des Kapitels wichtige Grundlagen vermittelt. Zunächst werden in Kapitel 4.2.1 grundlegende Begriffe und Zusammenhänge der Bildgebung mittels eines Kamerasystems eingeführt. Dabei wird auf die Projektion von Punkten im dreidimensionalen Raum auf die zweidimensionale Bildebene einer Kamera eingegangen und die mathematischen Zusammenhänge erklärt. Kapitel 4.2.2 und 4.2.3 widmen sich zwei wichtigen Problemstellungen der Mathematik, die bei der entwickelten Kalibrationsroutine eine zentrale Rolle spielen: Einen Sonderfall einer Ausgleichsrechnung zur Bestimmung eines zweidimensionalen Kreises, der einen gegebenen Satz von Punkten bestmöglich beschreibt, sowie der Koordinatentransformation zur Überführung von Punktkoordinaten eines Koordinatensystems in ein anderes. Das zur Interpolation von diskreten Werten, welche in einem 2D-Gitter vorliegen, einsetzbare Verfahren der bilinearen Interpolation wird in Kapitel 4.2.4 umrissen.

4.2.1. Das Kameramodell der Computer Vision

In der Computer Vision wird ein Kamerasytem bestehend aus einer Linsenoptik und einem Bildsensor vereinfacht durch ein Lochkameramodell approximiert [48, S. 45-49]. Die wichtigsten Begriffe und Parameter dieses Modells sind in Abbildung 4.1 dargestellt.

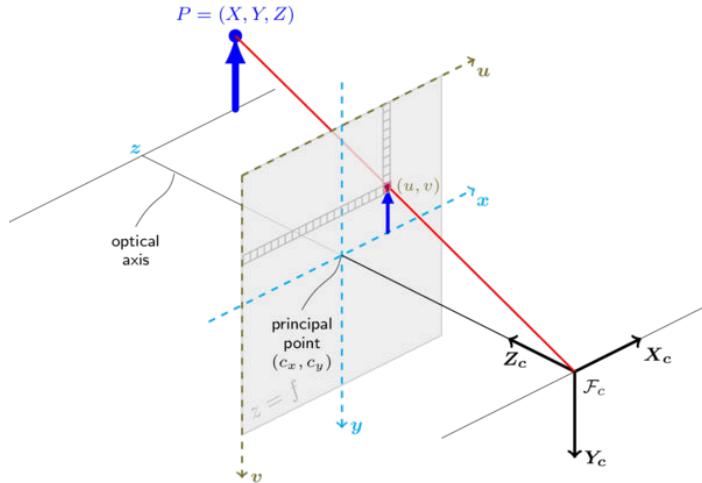


Abbildung 4.1.: Das Lochkamera-Modell [64]

Die zentrale Aufgabe eines Kamerasytems ist die Erzeugung von Bildaufnahmen. Dazu wird Licht, das von Lichtquellen emittiert oder Oberflächen reflektiert wurde, durch eine Optik auf einen fotosensitiven Sensor geführt. Vereinfacht kann dieser Vorgang auch als die Projektion von Punkten im dreidimensionalen Raum auf eine zweidimensionale Bildebene betrachtet werden. Ein Punkt $p \in \mathbb{R}^3$ in einem kamerazentrierten Koordinatensystem F_c

$$p = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

wird dabei auf einen Punkt $p_{\text{Bild}} \in \mathbb{P}^2$ in homogenen Bildkoordinaten

$$p'_{\text{Bild}} = \begin{pmatrix} u' \\ v' \\ f \end{pmatrix}$$

durch eine perspektivische Transformation mit der reellen Matrix $K \in \mathbb{R}^{3 \times 3}$

$$K = \begin{pmatrix} f_x & s & u_x \\ 0 & f_y & u_y \\ 0 & 0 & w \end{pmatrix} \quad (4.1)$$

gemäß folgender Gleichung überführt

$$p_{\text{Bild}} = K \cdot p \quad (4.2)$$

Die Matrix K wird als *Kamerakalibrationsmatrix* bezeichnet. Sie enthält die intrinsischen Kameraparameter und wird oftmals als obere Dreiecksmatrix gemäß Gleichung (4.1) mit den beiden unabhängigen Brennweiten f_x und f_y in X- bzw. Y-Richtung, dem Verzerrungsfaktor s zwischen X- und Y-Achse und den Koordinaten des Bildhauptpunktes u_x und u_y in Pixelkoordinaten dargestellt. Als Bildhauptpunkt (*principal point*) wird der Punkt bezeichnet, an dem die optische Achse (*optical axis*) die Bildebene durchdringt (vgl. Abbildung 4.1).

Um die Pixelkoordinaten u und v in der Bildebene zu erhalten, müssen die berechneten, homogenen Koordinaten anschließend noch normiert werden. Dies geschieht durch Division mit dem Skalierungsfaktor w . Gleichung (4.2) lässt sich somit umformulieren zu

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{w} \cdot K \cdot p \quad (4.3)$$

Bei bekannter Entfernung eines Bildpunktes von der Bildebene z ist die durch Gleichung (4.3) beschriebene Transformation mittels der invertierten Kameramatrix K^{-1} umkehrbar, wie folgende Gleichung zeigt

$$p = K^{-1} \cdot z \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \quad (4.4)$$

Die Komponenten der Matrix K werden im Zuge der Kamerakalibration bestimmt. Ein solches Verfahren wird von Zhang in seiner Arbeit vorgestellt [63].

4.2.2. Kreis-Fitting

Unter dem Begriff des *Kreis-Fittings*, das auch als Kreisausgleichsproblem bekannt ist, versteht man die Suche nach einem Kreis, der eine gegebene Punktemenge möglichst gut beschreibt. Für das Verständnis des im weiteren Verlauf dieses Kapitels vorgestellten Verfahrens genügt die Kenntnis eines zweidimensionalen Kreis-Fittings. An dieser Stelle wird daher lediglich die auf zweidimensionale Koordinaten reduzierte Formulierung des Problems vorgestellt.

Sei eine Punktemenge $P = \{p_1, p_2, \dots, p_n\}$ bestehend aus $n \in \mathbb{N}$ Punkten des \mathbb{R}^2 mit den Koordinaten

$$p_1 = \begin{pmatrix} p_{1,x} \\ p_{1,y} \end{pmatrix}, \quad p_2 = \begin{pmatrix} p_{2,x} \\ p_{2,y} \end{pmatrix}, \quad \dots, \quad p_n = \begin{pmatrix} p_{n,x} \\ p_{n,y} \end{pmatrix}$$

gegeben. Dann lässt sich die Suche nach einem Kreis mit Mittelpunkt m

$$m = \begin{pmatrix} m_x \\ m_y \end{pmatrix}, \quad m \in \mathbb{R}^2$$

und Radius $r \in \mathbb{R}^+$ als Minimierungsproblem formulieren zu

$$\min \sum_{i=1}^n \left| r - \sqrt{(p_{i,x} - m_x)^2 + (p_{i,y} - m_y)^2} \right|$$

Zur Lösung eines solchen Minimierungsproblems kann etwa das Levenberg-Marquardt Verfahren [65] eingesetzt werden. Eine Implementierung dieses Verfahrens findet sich auch in der weit verbreiteten Bibliothek für numerische lineare Algebra *LAPACK* [66].

4.2.3. Koordinatentransformation

Bei der Koordinatentransformation werden die Koordinaten eines Punktes in einem Koordinatensystem in Koordinaten eines anderen Koordinatensystems überführt. Dabei wird zwischen verschiedenen Transformationen entsprechend der vorhandenen Freiheitsgrade unterschieden [48, S. 33-42]. Im Weiteren werden lediglich auf affine Transformationen im zweidimensionalen Raum eingegangen. Affine Transformationen erlauben eine Verschiebung, Rotation, Skalierung und Verzerrung der Koordinaten. Im zweidimensionalen Raum entspricht dies sechs und im dreidimensionalen Raum 12 Freiheitsgraden.

Der Ausdruck von Punkten in homogenen Koordinaten erlaubt eine einfache Formulierung dieser Transformationen als Matrixmultiplikation. Ein zweidimensionaler Punkt $x = (x, y) \in \mathbb{R}^2$ lässt sich in homogenen Koordinaten darstellen als $\tilde{x} = (\tilde{x}, \tilde{y}, \tilde{w}) \in \mathbb{P}^2$. $\mathbb{P}^2 = \mathbb{R}^3 - (0, 0, 0)$ wird dabei als der zweidimensionale projektive Vektorraum bezeichnet. In diesem Raum werden alle Punkte, die sich lediglich durch einen Skalierungsfaktor unterscheiden, als identisch betrachtet. Durch die Verwendung von homogenen Koordinaten lässt sich die affine Transformation von Punkten im zweidimensionalen Raum mittels der Transformationsmatrix $T_{2D} \in \mathbb{R}^{3 \times 3}$

$$T_{2D} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{pmatrix}$$

formulieren. Dabei wird ein Punkt $p_1 \in \mathbb{P}^2$ in homogenen Koordinaten

$$p_1 = \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

auf den ebenfalls in homogenen Koordinaten dargestellten Punkt $p_2 \in \mathbb{P}^2$

$$p_2 = \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix}$$

gemäß folgender Gleichung überführt

$$p_2 = T_{2D} \cdot p_1 \quad (4.5)$$

Eine Transformation in umgekehrter Richtung ist mit der inversen Transformationsmatrix T_{2D}^{-1} ebenfalls möglich

$$p_1 = T_{2D}^{-1} \cdot p_2$$

Durch den Einsatz von homogenen Koordinaten lässt sich eine Verkettung von zwei Transformationen als Matrixmultiplikation ausdrücken.

Bei vielen technischen Problemen liegen Punkte in unterschiedlichen Koordinatensystemen vor und es wird eine Transformationsmatrix gesucht, welche die Punkte möglichst gut aufeinander abbildet. Es wird im Weiteren davon ausgegangen, dass die Zuordnungen jedes Punktes $p_{1,i} \in \mathbb{P}^2$

$$p_{1,i} = \begin{pmatrix} x_{1,i} \\ y_{1,i} \\ 1 \end{pmatrix}$$

in einem Koordinatensystem S_1 zu einem entsprechenden Punkt $p_{2,i} \in \mathbb{P}^2$

$$p_{2,i} = \begin{pmatrix} x_{2,i} \\ y_{2,i} \\ 1 \end{pmatrix}$$

bekannt sind. Jede Punktentsprechung liefert dabei gemäß Gleichung (4.5) zwei Gleichungen

$$x_{2,i} = a \cdot x_{1,i} + b \cdot y_{1,i} + c$$

$$y_{2,i} = d \cdot x_{1,i} + e \cdot y_{1,i} + f$$

Um die sechs unbekannten Parameter der gesuchten Transformationsmatrix $T_{2D,S_1,S_2} \in \mathbb{R}^{3 \times 3}$ eindeutig zu bestimmen, werden daher wenigstens drei Punktentsprechungen benötigt. Die Berechnung der Transformationsmatrix lässt sich dann mit $n \in \mathbb{N}^+$ Punktentsprechungen als Lösung des folgenden Gleichungssystems der Form $A \cdot x = b$, wobei $A \in \mathbb{R}^{2n \times 6}$, $x \in \mathbb{R}^6$ und $b \in \mathbb{R}^{2n}$ formulieren:

$$\begin{pmatrix} x_{1,1} & y_{1,1} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{1,1} & y_{1,1} & 1 \\ x_{1,2} & y_{1,2} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{1,2} & y_{1,2} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{1,n} & y_{1,n} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{1,n} & y_{1,n} & 1 \end{pmatrix} \cdot \begin{pmatrix} t_{11} \\ t_{12} \\ t_{13} \\ t_{21} \\ t_{22} \\ t_{23} \end{pmatrix} = \begin{pmatrix} x_{2,1} \\ y_{2,1} \\ x_{2,2} \\ y_{2,2} \\ \vdots \\ x_{2,n} \\ y_{2,n} \end{pmatrix} \quad (4.6)$$

Bei drei nicht linear abhängigen Punktentsprechungen liefert das Gleichungssystem aus (4.6) eine exakte Lösung. In der Praxis sind die Punktkoordinaten jedoch häufig fehlerbehaftet. In einem solchen Fall wird durch die Verwendung von $n \geq 3$ Punktentsprechungen ein überbestimmtes Gleichungssystem geschaffen. Dieses wird anschließend durch Minimierung der Fehlerquadrate entsprechend folgender Gleichung gelöst

$$\min |b - A \cdot x|$$

Einige auf Geschwindigkeit optimierte Algorithmen zur Lösung dieses Problems finden sich in der *LAPACK* Bibliothek [66]. Eines dieser Verfahren ist das unter der Bezeichnung *GELSD* implementierte Verfahren. Dieses nutzt die Eigenwertzerlegung der Matrix A und eine *Divide and Conquer* Methode zur Lösungsfindung.

Eine affine Transformation lässt sich auch für Punkte im dreidimensionalen Raum formulieren. Dabei besitzt eine solche Transformation 12 Freiheitsgrade und kann durch eine Matrix $T_{3D} \in \mathbb{R}^{4 \times 4}$

$$T_{3D} = \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

dargestellt werden. Auch hierbei wird die Darstellung der zu transformierenden Punkte in homogenen Koordinaten genutzt.

Das beschriebene Verfahren zur Bestimmung einer Transformationsmatrix aus Punktentsprechungen im zweidimensionalen Raum lässt sich auch auf den dreidimensionalen Fall im \mathbb{R}^3 übertragen. Jede Punktentsprechung im dreidimensionalen Vektorraum liefert dabei drei Gleichungen. Für eine eindeutige Bestimmung der Transformationsmatrix sind daher in diesem Fall wenigstens vier Punktentsprechungen zur Bestimmung der 12 Parameter notwendig. Die Bildungsvorschrift für das Lineare Gleichungssystem zur Bestimmung der unbekannten Parameter der Transformationsmatrix T_{3D,R_1,R_2} zur Überführung der Punkte $q_{1,i} \in \mathbb{P}^3$

$$q_{1,i} = \begin{pmatrix} x_{1,i} \\ y_{1,i} \\ z_{1,i} \\ 1 \end{pmatrix}$$

des dreidimensionalen Koordinatensystems R_1 in Punkte $q_{2,i} \in \mathbb{P}^3$

$$q_{2,i} = \begin{pmatrix} x_{2,i} \\ y_{2,i} \\ z_{2,i} \\ 1 \end{pmatrix}$$

des Koordinatensystems R_2 mittels $m \in \mathbb{N}^+$ Punktentsprechungen lautet wie folgt

$$\left(\begin{array}{ccccccccccccc} x_{1,i} & y_{1,i} & z_{1,i} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_{1,i} & y_{1,i} & z_{1,i} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_{1,i} & y_{1,i} & z_{1,i} & 1 & 0 \\ x_{1,i+1} & y_{1,i+1} & z_{1,i+1} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_{1,i+1} & y_{1,i+1} & z_{1,i+1} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_{1,i+1} & y_{1,i+1} & z_{1,i+1} & 1 & 0 \\ \vdots & \vdots \\ x_{1,m} & y_{1,m} & z_{1,m} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_{1,m} & y_{1,m} & z_{1,m} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_{1,m} & y_{1,m} & z_{1,m} & 1 \end{array} \right) \cdot \begin{pmatrix} t_{11} \\ t_{12} \\ t_{13} \\ t_{14} \\ t_{21} \\ t_{22} \\ t_{23} \\ t_{24} \\ t_{31} \\ t_{32} \\ t_{33} \\ t_{34} \end{pmatrix} = \begin{pmatrix} x_{2,i} \\ y_{2,i} \\ z_{2,i} \\ x_{2,i+1} \\ y_{2,i+1} \\ z_{2,i+1} \\ x_{2,m} \\ y_{2,m} \\ z_{2,m} \end{pmatrix} \quad (4.7)$$

4.2.4. Bilineare Interpolation

Die bilineare Interpolation ist eine Erweiterung der linearen Interpolation für Funktionen mit zwei Variablen, deren Werte in einem diskreten, zweidimensionalen Gitter vorliegen [67, S. 132 f.].

Sei der Wert einer Funktion $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ für einen Punkt $p = (x_0, y_0)$ im \mathbb{R}^2 zu bestimmen und die den Punkt umgebenden Rasterpunkte $P_{11} = (x_1, y_1)$, $P_{12} = (x_1, y_2)$, $P_{21} = (x_2, y_1)$ und $P_{22} = (x_2, y_2)$ mit $x_1 \leq x \leq x_2$ und $y_1 \leq y \leq y_2$ gegeben. Bei der bilinearen Interpolation werden zunächst die Funktionswerte der Rasterpunkte entlang einer zu Beginn definierten Dimension interpoliert. In diesem Beispiel wird hierfür die erste Dimension gewählt. Die Funktionswerte an den beiden Stützpunkten $R_1 = (x_0, y_1)$ und $R_2 = (x_0, y_2)$ werden dann entsprechend der folgenden Gleichungen interpoliert

$$f(R_1) \approx \frac{x_2 - x_0}{x_2 - x_1} f(P_{11}) + \frac{x_0 - x_1}{x_2 - x_1} f(P_{21})$$

$$f(R_2) \approx \frac{x_2 - x_0}{x_2 - x_1} f(P_{12}) + \frac{x_0 - x_1}{x_2 - x_1} f(P_{22})$$

Durch eine erneute Interpolation der Werte an den berechneten Stützpunkten $f(R_1)$ und $f(R_2)$ entlang der zweiten Dimension kann dann der gesuchte Funktionswert für den Punkt p entsprechend Gleichung berechnet werden zu

$$f(p) \approx \frac{y_2 - y_0}{y_2 - y_1} f(R_1) + \frac{y_0 - y_1}{y_2 - y_1} f(R_2)$$

Dieses Verfahren kann beispielsweise genutzt werden, um Farbwerte an beliebigen Positionen, unabhängig des Pixelrasters, in einer Bildaufnahme zu bestimmen.

4.3. Eigener Ansatz

Die von Weck [60] vorgestellten Methoden zur Kalibrierung und Vermessung von Werkzeugmaschinen liefern sehr exakte Ergebnisse und erlauben eine entsprechend gute Kalibrierung. Gleichzeitig sind sie jedoch durch die benötigten Messmittel sehr kostenintensiv. Um einen einfachen Nachbau der entwickelten Kontakthardware im Sinne des Open Hardware-Gedankens zu ermöglichen, wurde ein sehr kostengünstiges und dabei dennoch präzises Verfahren zur Kalibration entwickelt. Kern der Kalibrierung ist ein künstlich geschaffenes Referenzkoordinatensystem in Form einer 3D-gedruckten Referenzplatte. In diese Platte sind mehrere metallische Zylinder eingelassen, deren Mittelpunkte als Referenzen bei der Kalibrierung dienen. Diese können sowohl von den Prüfeinheiten, als auch durch das Kamerasytem einfach und vor allem sehr präzise erfasst werden.

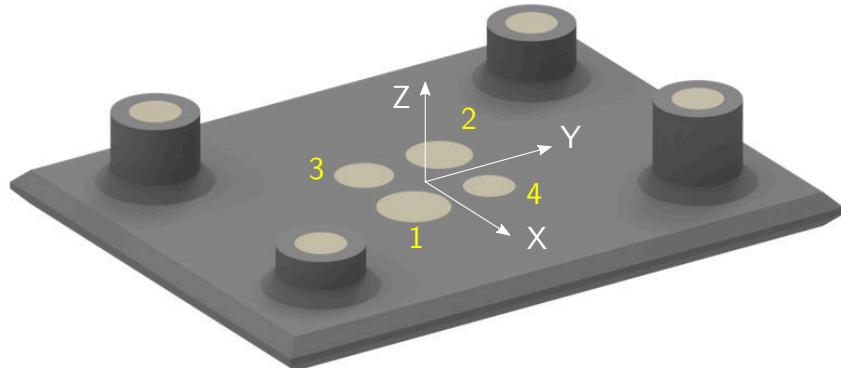


Abbildung 4.2.: Rendering der entworfenen Referenzplatte mit eingezeichnetem Referenzkoordinatensystem und IDs der Referenzpins

Abbildung 4.2 zeigt den Aufbau dieser Referenzplatte. In die 3D-gedruckte Grundplatte aus schwarzem Kunststoff sind mehrere metallische Referenzzyliner mit variierenden Durchmessern in unterschiedlichen Höhen eingelassen. Der Ursprung des künstlichen Referenzsystems liegt im Zentrum der Platte. Die Orientierung des Koordinatensystems wird durch die vier mittigen Zylinder mit jeweils unterschiedlichem Durchmesser von 6 bis 10 mm definiert. Jedem Referenzzyliner wurde eine eindeutige ID zugeordnet, welche in Abbildung 4.2 in gelber Farbe

eingezeichnet sind. Am Rande der Platte befinden sich vier weitere Zylinder in unterschiedlichen Höhen von 6 bis 12 mm mit einem Durchmesser von jeweils 7 mm. Die Seiten der Referenzplatte sind angeschrägt, so dass sie von der entworfenen Platinenhalterung der Kontaktierhardware aufgenommen werden kann.

Die eingelassenen Zylinder können durch das eingesetzte Kamerasystem aufgrund der Farbunterschiede sehr einfach und exakt erfasst werden. Gleichzeitig können diese anhand ihrer elektrischen Leitfähigkeit durch die Prüfspitzen sehr leicht von dem umgebenden, isolierenden Kunststoff unterschieden werden. Durch die Bestimmung mehrerer Randpunkte der Referenzzyliner können anschließend mit dem in Kapitel 4.2.2 beschriebenen Verfahren zum Kreis-Fitting die Mittelpunkte berechnet werden. Neben den Mittelpunktkoordinaten liegen als Ergebnis des Kreis-Fittings zusätzlich die Radien der Referenzzyliner vor, durch welche diese eindeutig identifiziert werden können. Als Ergebnis liegen die Referenzpunkte der Kalibrierungsplatte in den lokalen Koordinatensystemen der Prüfeinheiten bzw. des Kamerasystems vor. Mit diesen Informationen können, wie in den nachfolgenden Kapiteln noch näher erläutert wird, Transformationen aus den lokalen Koordinatensystemen der einzelnen Einheiten in das künstlich geschaffene Referenzsystem berechnet werden. Die Transformationen zwischen den Koordinatensystemen werden dabei durch affine Transformationen ausgedrückt. Diese erlauben neben der Translation, Rotation und Skalierung auch Verzerrungen der Koordinatenachsen¹. Es lassen sich somit auch Fehler der Rechtwinkligkeit zwischen den einzelnen Achsen der Koordinatensysteme abbilden. Diese können etwa durch eine ungenaue Montage der Schlittensysteme entstehen. Mittels der so bestimmten Transformationen können beliebige Punkte aus den lokalen Koordinatensystemen der Kontaktiereinheiten, mit einem Zwischenschritt über das künstlich geschaffene Referenzsystem, in das einer beliebigen, anderen Einheit umgerechnet werden.

Die so bestimmten Transformationen geben außerdem Aufschluss über etwaige Montage oder Fertigungsungenauigkeiten der Einheiten, wie etwa Winkel- und im Besonderen Rechtwinkligkeitsfehler zwischen den einzelnen Achsen einer Einheit. Diese können anschließend mechanisch behoben oder durch Software-Algorithmen kompensiert werden.

In den nachfolgenden Kapiteln wird der Ablauf zur Kalibrierung der Kontaktiereinheiten und des eingesetzten, fahrenden Kamerasystems im Detail erklärt. Ein besonderer Fokus liegt auf der Darstellung der mathematischen Zusammenhänge zur Berechnung der Transformationen zwischen den lokalen Koordinatensystemen der Kontaktiereinheiten und Bildkoordinaten des Kamerasystems.

4.4. Kalibrierung der Kontaktiereinheiten

Aufgrund der elektrischen Leitfähigkeit der Referenzzyliner in der Referenzplatte und der isolierenden Eigenschaft des 3D-gedruckten Kunststoffes der umgebenden Platte können diese sehr genau durch die Prüfspitzen ausgemessen werden. Dazu wird die Prüfspitze mit einem

¹vgl. Kapitel 4.2.3

als *Pull-Up* parametrierten GPIO-Pin der verwendeten Steuerplatten und die Referenzzyliner selbst mit *Ground* verbunden. An dem GPIO-Pin liegt somit standardmäßig ein *HIGH*-Pegel an. Trifft die Prüfspitze nun aber auf einen der Referenzzyliner, so wird der Pegel des GPIO-Pins auf *LOW* gezogen. Auf diese Weise ist eine Unterscheidung zwischen einem Referenzzyliner und dem umgebenden Kunststoff der Referenzplatte möglich und kann softwaretechnisch sehr einfach ausgewertet werden.

Zur möglichst exakten Bestimmung der Mittelpunkte der Referenzzyliner wurde ein eigenständiger Zyklus auf der Steuerplatine implementiert. Dieser kann über den *M*-Befehl *M137* aktiviert werden. Der Zyklus setzt die beschriebene Verschaltung eines bestimmten GPIO-Pins voraus. Der Zyklus bestimmt mittels binärer Suche vier Punkte auf dem Umfang des Referenzzyliners. Durch ein anschließendes Kreisfitting wird das Zentrum des Zylinders berechnet.

Abbildung 4.3 zeigt vereinfacht das Verfahren zur Bestimmung der vier Kantenpunkte. Nach dem Start des Zyklus wird die Prüfspitze solange abgesenkt, bis die metallische Oberfläche des Referenzzyliners kontaktiert wird (Abbildung 4.3, links). Die Spitze wird daraufhin wieder angehoben, um einen Schritt der Länge $s \in \mathbb{R}^+$ in Richtung der positiven X-Achse bewegt und anschließend erneut abgesenkt. Diese Schritte werden solange wiederholt bis sich die Prüfspitze außerhalb des Referenzzyliners befindet. Daraufhin wird die Schrittweite s halbiert und gleichzeitig die Schrittrichtung invertiert (Abbildung 4.3, Mitte). Dieses einer binären Suche entsprechende Verfahren wird solange fortgeführt, bis eine voreingestellte Mindestschrittweite unterschritten ist. Die aktuelle Position der Prüfspitze wird dann als Randpunkt gespeichert. Durch die Mindestschrittweite wird die Genauigkeit der Punktkoordinaten maßgeblich beeinflusst. In der aktuellen Implementierung wird eine Mindestschrittweite von 0,005 mm festgelegt, was dem Doppelten der Mikroschrittauflösung der Antriebe in der X- und Y-Achse der Hardwareplattform entspricht.

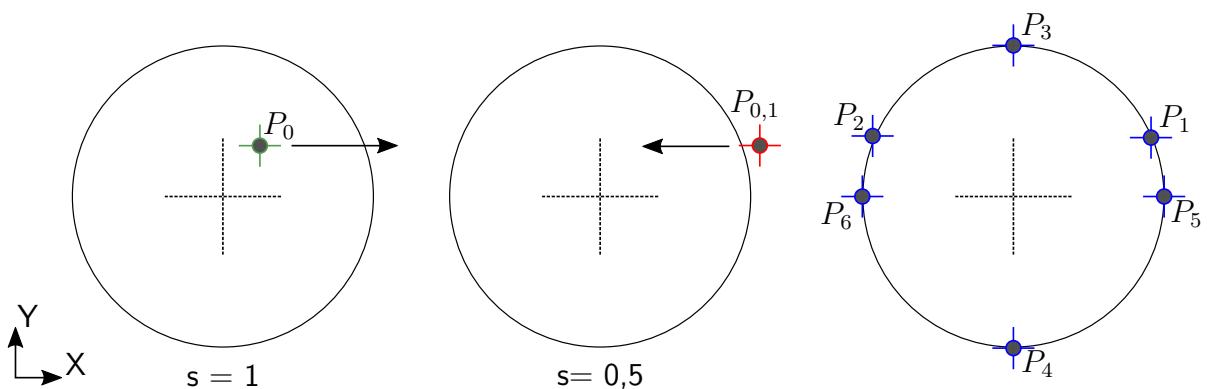


Abbildung 4.3.: Schritte bei der Bestimmung des Zentrums eines Referenzzyliners mittels elektrischer Kontaktierung

Um durch das nachfolgende Kreis-Fitting möglichst exakt den Mittelpunkt der Referenzzyliner zu bestimmen und um den Einfluss von Messfehlern in den Koordinaten der Randpunkte zu minimieren, werden durch den Zyklus kreuzförmig die am weitesten voneinander entfer-

ten Randpunkte bestimmt. Diesen Sachverhalt zeigt die rechte schematische Darstellung in Abbildung 4.3: Nachdem der Randpunkt P_1 durch den Zyklus bestimmt wurde, wird der entsprechend gegenüberliegende Punkt P_2 auf der selben Achse ertastet. Es wird die mittlere X-Koordinate aus den beiden Punkten als vorläufiger Mittelpunkt des Referenzzyinders gebildet. Entlang einer durch diesen Punkt verlaufenden Geraden in Y-Richtung werden anschließend die Randpunkte P_3 und P_4 angetastet. Abschließend erfolgt eine erneute Bestimmung der Randpunkte in X-Richtung, nachdem aus den Punkten P_3 und P_4 die Mittelpunktskoordinate entlang der Y-Achse des Referenzzyinders grob durch Mittelwertbildung berechnet wurde. Von den insgesamt sechs bestimmten Randpunkten werden nur die Punkte P_3 , P_4 , P_5 und P_6 mit dem größtmöglichen Abstand zueinander für die weiteren Berechnungen verwendet.

Die Mittelpunkte der Referenzzyinder werden anschließend mittels dem in Kapitel 4.2.2 beschriebenen Kreis-Fitting Algorithmus berechnet. Durch die unterschiedlich großen Durchmesser können die gemessenen Mittelpunkte der Referenzzyinder eindeutig Punkten des künstlichen Referenzsystems zugeordnet werden. Diesen Zusammenhang zeigt Abbildung 4.4.

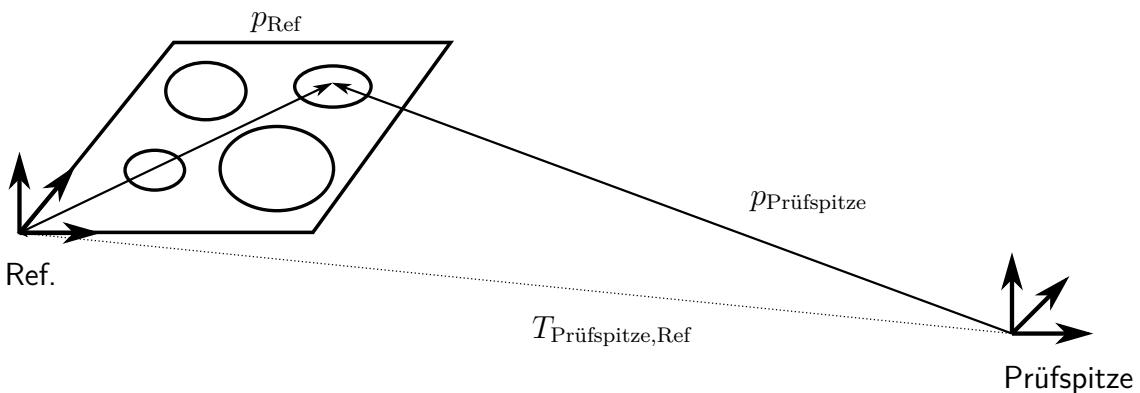


Abbildung 4.4.: Zusammenhang zwischen Punkten des Referenzsystems und dem Koordinatensystem einer Prüfspitze

Die vier Referenzzyinder in der Mitte der Referenzplatte befinden sich alle in einer Ebene. Um auch die Abbildung der Z-Achsen zu bestimmen, werden zusätzlich die vier Referenzzyinder am Rand der Referenzplatte mit dem bereits beschriebenen Verfahren angetastet. Somit liegen insgesamt acht Punktpaare aus Punkten des Referenzsystems und des lokalen Koordinatensystems der Kontaktiereinheit vor.

Damit nun für beliebige Punkte im Referenzsystem die zugehörigen Koordinaten im System der Prüfspitze berechnet werden können, wird die Transformationsmatrix $T_{\text{Ref}, \text{Prufspitze}}$ bestimmt. Dazu wird das in Kapitel 4.2.3 vorgestellte Verfahren zur Berechnung der Transformationsmatrix anhand mehrerer Punktentsprechungen im dreidimensionalen Raum eingesetzt. Das Gleichungssystem mit 12 Unbekannten ist durch die Verwendung aller acht gemessenen Punkte und den damit definierten 16 Gleichungen überbestimmt (vgl. Gleichung (4.7)). Die Formulierung als überbestimmtes Gleichungssystem erlaubt an dieser Stelle die Minimierung des Einflusses von Messungenauigkeiten bei der Bestimmung der finalen Transformationsmatrix.

Mit der so gewonnenen Transformationsmatrix können mittels der folgenden Gleichung beliebige Punkte $p_{\text{Ref}} \in \mathbb{P}^3$ aus dem Referenzkoordinatensystem in entsprechende Punkte des Systems der Prüfeinheit $p_{\text{Prüfspitze}} \in \mathbb{P}^3$ überführt werden.

$$p_{\text{Prüfspitze}} = T_{\text{Ref}, \text{Prüfspitze}} \cdot p_{\text{Ref}} \quad (4.8)$$

Die Punkte werden dabei durch die in Kapitel 4.2.3 beschriebenen homogenen Koordinaten dargestellt. Eine Transformation in umgekehrter Richtung ist mit der invertierten Transformationsmatrix $T_{\text{Prüfspitze}, \text{Ref}}^{-1}$ ebenfalls möglich:

$$p_{\text{Ref}} = T_{\text{Ref}, \text{Prüfspitze}}^{-1} \cdot p_{\text{Prüfspitze}}$$

4.5. Kalibrierung des bewegten Kamerasytems

In dem aufgebauten Prototyp wird ein einfaches USB-Mikroskop als Kamerasytem zur Bildgebung eingesetzt. Dieses ist wiederum auf einem in der X- und Y-Achse frei verfahrbaren Schlitten zusammen mit einer zusätzlich in Z-Richtung beweglichen Prüfspitze montiert. Der Aufbau ist in Abbildung 4.5 zu sehen.

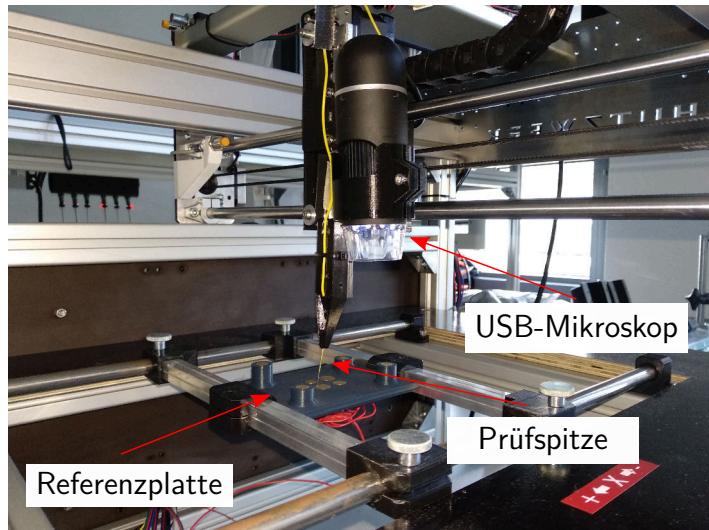


Abbildung 4.5.: Fahrendes Kamerasytem im aufgebauten Prototyp

Für die Kalibrierung dieses Teilsystems wird zunächst eine Kalibrierung der Kamera gemäß dem von Zhang [63] vorgestellten Verfahren zur Bestimmung der intrinsischen Kameraparameter durchgeführt. Der Algorithmus ist bereits in der Computer Vision Bibliothek [29] implementiert und konnte an dieser Stelle genutzt werden. Als Ergebnis des Algorithmus werden sowohl die Kameramatrix K als auch die nicht linearen Linsenparameter k_1, \dots, k_6 ausgegeben. Letztere beschreiben eine kissen- oder tonnenförmige Verzeichnung bzw. Verzerrung des Kamerabilds durch die eingesetzte Optik [5, S. 61 f.]. Da das eingesetzte USB-Mikroskop jedoch keine

merkliche Linsenverzeichnung aufweist, wurden die nichtlinearen Linsenparameter in die weiteren Berechnungen nicht miteinbezogen.

Das Kamerasytem liefert Bildaufnahmen des untersuchten PCBs. Da die Analyse des PCBs primär auf diesen Bildaufnahmen durchgeführt wird, muss eine Möglichkeit geschaffen werden, aus Pixelkoordinaten der Bildaufnahmen entsprechende Koordinaten für die Prüfspitzen zu berechnen. Um dies zu ermöglichen, werden Bildkoordinaten in Punkte des in Kapitel 4.3 vorgestellten Referenzsystems überführt. Zur Berechnung der Transformationsmatrizen werden ebenfalls die metallischen Zylinder in der Referenzplatte genutzt. Aus dem Referenzsystem können die Punkte dann in Koordinaten einer beliebigen Prüfeinheit mittels der in Kapitel 4.4 beschriebenen Transformationsmatrizen umgerechnet werden.

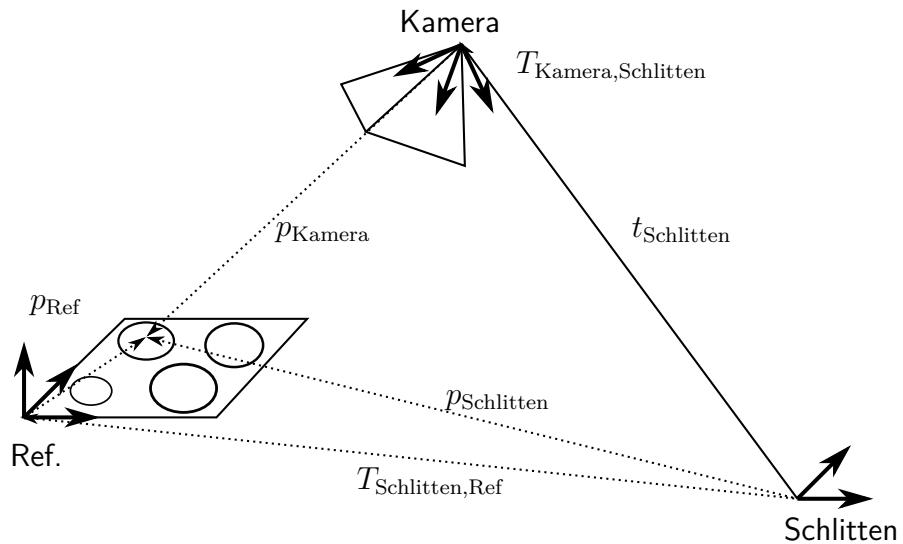


Abbildung 4.6.: Involvierte Koordinatensysteme bei der Kalibrierung des bewegten Kamerasytems

Abbildung 4.6 zeigt die bei der Bestimmung der Transformation aus dem Kamerasytem in das Referenzsystem beteiligten Koordinatensysteme und Transformationen. Im ersten Schritt werden die Referenzzyliner im Kamerabild lokalisiert. Anschließend erfolgt mittels der invertierten Kamerakalibrationsmatrix eine Rückprojektion der Bildpunkte in das Kamerakoordinatensystem. Durch mehrere Aufnahmen mit unterschiedlichen Positionen des Kameraschlittens $t_{\text{Schlitten}}$ wird die Transformationsmatrix $T_{\text{Kamera},\text{Schlitten}}$ aus dem Kamerasytem in das zweidimensionale Schlittensystem bestimmt. Zuletzt wird durch die bekannten Abbildungen der Referenzpunkte in dem Schlittensystem die Transformation von Koordinaten des Schlittensystems in das Referenzsystem mittels der Transformationsmatrix $T_{\text{Schlitten},\text{Ref}}$ bestimmt. Die genannten Schritte und die genauen mathematischen Zusammenhänge werden in den folgenden Abschnitten näher ausgeführt.

Abbildung 4.7a zeigt, dass die Referenzzyliner im Kamerabild sehr leicht erkannt werden können. Die goldene Farbe der aus Messing gefertigten Zylinder zeichnet sich deutlich von dem umliegenden schwarzen Kunststoff der Referenzplatine ab. Für eine automatische Erkennung

der Referenzzyliner genügt daher ein einfaches globales Farbschwellwertverfahren². Um die Robustheit gegenüber schwankenden Lichtverhältnissen zu erhöhen, wird die Schwellwertbildung im HSV-Farbraum durchgeführt. Die Eingrenzung des Farbbereichs erfolgt dabei primär auf dem Kanal des Farbwertes H . Ein Beispiel für das Resultat des binarisierten Kamerabildes ist in Abbildung 4.7b zu sehen. In der so erzeugten binären Maske wird anschließend eine Bestimmung von Zusammenhangskomponenten gefolgt von einer Konturfindung durchgeführt³. Als Ergebnis liegen die Konturpixel auf dem Rand der Referenzzyliner vor. Die Zentren der einzelnen Zylinder werden ebenfalls durch das bereits bei der Kalibrierung der Prüfeinheiten eingesetzte Kreis-Fitting Verfahren berechnet. Dieses liefert die Mittelpunkte und jeweiligen Radien der Referenzzyliner in Bildkoordinaten des Kamerasytems. Die erkannten Referenzzylinderkonturen sind in Abbildung 4.7c als rote Kreise dargestellt. Die erkannten Zuordnungen der Konturen zu den entsprechenden Referenzpins sind darin ebenfalls durch die eingezeichneten Referenzpin-IDs visualisiert. Anhand der unterschiedlichen Radien können auch hier die detektierten Referenzpunkte eindeutig Koordinaten des Referenzsystems zugeordnet werden.

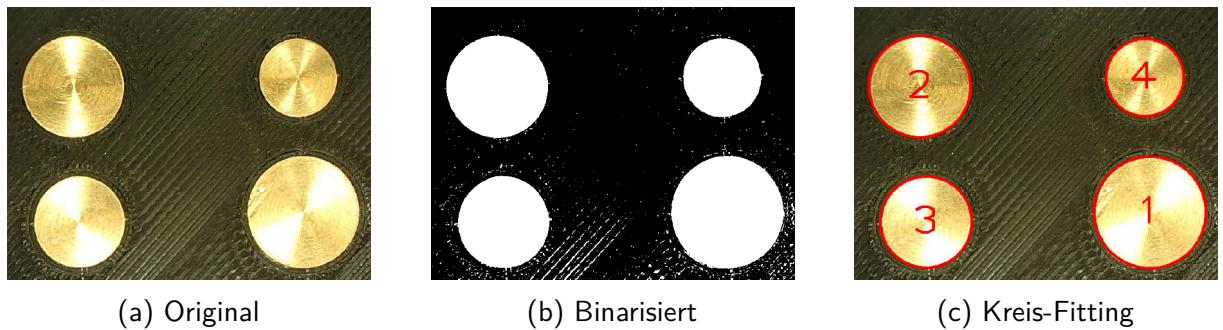


Abbildung 4.7.: Erkennung der Referenzzyliner im Kamerabild

Für eine Rückprojektion von Punkten aus der Bildebene in das Kamerakoordinatensystem werden neben den Koordinaten der Bildebene zusätzlich die realen Entfernung der Bildpunkte bezüglich der Bildebene in Z-Richtung des Kamerasytems benötigt. Durch Kenntnis der Abstände der Referenzzyliner zueinander können diese unter Zuhilfenahme der inversen Kameramatrix K^{-1} wie folgt berechnet werden: Für die nachfolgenden Berechnungen wird angenommen, dass die Kamera und damit auch die Bildebene parallel zur Oberfläche der Referenzplatte ausgerichtet sind. Seien zwei Punkte auf der Referenzplatte p_1 und p_2

$$p_1 = \begin{pmatrix} x_1 \\ y_1 \\ z \end{pmatrix}, \quad p_2 = \begin{pmatrix} x_2 \\ y_2 \\ z \end{pmatrix} \quad p_1, p_2 \in \mathbb{R}^3$$

mit derselben unbekannten Entfernung z zur Bildebene gegeben. Die Distanz $d \in \mathbb{R}^+$ zwischen p_1 und p_2 auf der Referenzplatte berechnet sich gemäß nachfolgender Gleichung zu

²siehe Kapitel 2.2.3

³siehe Kapitel 2.2.8

$$\begin{aligned} d &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z - z)^2} \\ &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \end{aligned} \quad (4.9)$$

Die Punkte p_1 und p_2 können mittels der inversen Kameramatrix K^{-1}

$$K^{-1} = \begin{pmatrix} k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,1} & k_{3,2} & k_{3,3} \end{pmatrix}$$

aus den entsprechenden Bildpunkten $p_{\text{Bild},1}$ und $p_{\text{Bild},2}$

$$p_{\text{Bild},1} = \begin{pmatrix} u_1 \\ v_1 \end{pmatrix}, \quad p_{\text{Bild},2} = \begin{pmatrix} u_2 \\ v_2 \end{pmatrix} \quad p_{\text{Bild},1}, p_{\text{Bild},2} \in \mathbb{R}^2$$

gemäß Gleichung(4.4) berechnet werden. Die Differenzen zwischen den Koordinaten der Punkte p_1 und p_2 können damit folgendermaßen ausgedrückt werden:

$$\begin{aligned} (x_2 - x_1) &= z \cdot [u_2 \cdot k_{1,1} + v_2 \cdot k_{1,2} + k_{1,3} - (u_1 \cdot k_{1,1} + v_1 \cdot k_{1,2} + k_{1,3})] \\ &= z \cdot [(u_2 - u_1) \cdot k_{1,1} + (v_2 - v_1) \cdot k_{1,2}] \\ &= z \cdot a \end{aligned} \quad (4.10)$$

$$\begin{aligned} (y_2 - y_1) &= z \cdot [u_2 \cdot k_{2,1} + v_2 \cdot k_{2,2} + k_{2,3} - (u_1 \cdot k_{2,1} + v_1 \cdot k_{2,2} + k_{2,3})] \\ &= z \cdot [(u_2 - u_1) \cdot k_{2,1} + (v_2 - v_1) \cdot k_{2,2}] \\ &= z \cdot b \end{aligned} \quad (4.11)$$

Einsetzen von Gleichung (4.10) und (4.11) in Gleichung (4.9) erlaubt die direkte Bestimmung der unbekannten Entfernung der Bildpunkte z gemäß der Vorschrift

$$z = \frac{d}{\sqrt{a^2 + b^2}} \quad (4.12)$$

Nachdem so die gefundenen Koordinaten der Referenzpunkte im Kamerasytem bestimmt wurden, erfolgt im nächsten Schritt die Berechnung der Transformationsmatrix zur Überführung der Punkte aus dem Kamerasytem in das Koordinatensystem des in X- und Y-Richtung fahrbaren Schlittensystems, auf dem die Kamera montiert ist. Es handelt sich dabei lediglich um eine zweidimensionale Transformation in der Ebene, da die Kamera nicht in Z-Richtung bewegt und damit keine Abbildung in dieser Dimension zwischen den Koordinatensystemen bestimmt werden kann. Entsprechend Abbildung 4.6 berechnet sich ein Punkt im System des fahrbaren Schlittens $p_{\text{Schlitten},i}$ in homogenen Koordinaten

$$p_{\text{Schlitten},i} = \begin{pmatrix} x_{\text{Schlitten},i} \\ y_{\text{Schlitten},i} \\ 1 \end{pmatrix} \quad p_{\text{Schlitten},i} \in \mathbb{P}^2$$

aus einem Punkt im Kamerakoordinatensystem in homogenen Koordinaten $p_{\text{Kamera},i}$

$$p_{\text{Kamera},i} = \begin{pmatrix} x_{\text{Kamera},i} \\ y_{\text{Kamera},i} \\ z \\ 1 \end{pmatrix} \quad p_{\text{Kamera},i} \in \mathbb{P}^3$$

unter Berücksichtigung der aktuellen Schlittenposition in homogenen Koordinaten

$$t_{\text{Schlitten}} = \begin{pmatrix} x_{t,\text{Schlitten}} \\ y_{t,\text{Schlitten}} \\ 1 \end{pmatrix} \quad t_{\text{Schlitten}} \in \mathbb{P}^2$$

und der zweidimensionalen Transformationsmatrix $T_{\text{Kamera},\text{Schlitten}}$

$$T_{\text{Kamera},\text{Schlitten}} = \begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \quad T_{\text{Kamera},\text{Schlitten}} \in \mathbb{R}^{3 \times 3}$$

nach folgender Gleichung

$$p_{\text{Schlitten},i} = t_{\text{Schlitten}} + T_{\text{Kamera},\text{Schlitten}} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot p_{\text{Kamera},i} \quad (4.13)$$

Neben den Koeffizienten der Transformationsmatrix sind weiterhin die Koordinaten der Referenzpunkte im System des Schlittensystems $p_{\text{Schlitten},i}$ unbekannt und müssen im Zuge der Kalibration berechnet werden. Bei $m \in \mathbb{N}$ Referenzpunkten sind somit insgesamt $6 + 2 \cdot m$ Parameter zu bestimmen. Jede Punktentsprechung liefert zwei Gleichungen. Bei $m = 4$ im Kamerabild sichtbaren Referenzpunkten sind somit wenigstens $n = 2$ Aufnahmen mit unterschiedlichen Schlittenpositionen notwendig, um eine eindeutige Lösung für die Transformationsmatrix und die Referenzpunkte im Schlittenkoordinatensystem zu erhalten. Um Rotationen und Verzerrungen in den Achsen des Schlittensystems möglichst gut zu erfassen, werden in der implementierten Kalibrierungsroutine insgesamt vier unterschiedliche Schlittenpositionen angefahren. Die Positionsänderungen des Schlittens sind dabei maximal gewählt, so dass alle Referenzzyliner noch vollständig in jeder Kameraaufnahme sichtbar sind. Wird die ursprüngliche Abbildungsgleichung (4.13) entsprechend folgender Gleichung umformuliert

$$T_{\text{Kamera},\text{Schlitten}} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot p_{\text{Kamera},i} - p_{\text{Schlitten},i} = -t_{\text{Schlitten}}$$

lässt sich die Bestimmung der Transformationsmatrix $T_{\text{Kamera},\text{Schlitten}}$ und der Koordinaten der Referenzpunkte im Schlittensystem als ein Gleichungssystem der Form

$$A \cdot x = b$$

mit $A \in \mathbb{R}^{2mn, 6+2m}$, $x \in \mathbb{R}^{6+2m}$ und $b \in \mathbb{R}^{2mn}$ darstellen. Die folgende Gleichung zeigt die Bildungsvorschrift für die benötigte Matrix A und die beiden Vektoren x und b für die Referenzpunkte $i = 1, \dots, n$ und Schlittenpositionen $j = 1, \dots, m$.

$$\begin{pmatrix} x_{\text{Kamera},1,1} & y_{\text{Kamera},1,1} & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & x_{\text{Kamera},1,1} & y_{\text{Kamera},1,1} & 0 & 1 & 0 & -1 & 0 & 0 & \dots & 0 & 0 \\ \dots & & & & & & & & & & & & \\ x_{\text{Kamera},n,1} & y_{\text{Kamera},n,1} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots & -1 & 0 \\ 0 & 0 & x_{\text{Kamera},n,1} & y_{\text{Kamera},n,1} & 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 & -1 \\ \dots & & & & & & & & & & & & \\ x_{\text{Kamera},2,1} & y_{\text{Kamera},2,1} & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & x_{\text{Kamera},2,1} & y_{\text{Kamera},2,1} & 0 & 1 & 0 & -1 & 0 & 0 & \dots & 0 & 0 \\ \dots & & & & & & & & & & & & \\ x_{\text{Kamera},n,m} & y_{\text{Kamera},n,m} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots & -1 & 0 \\ 0 & 0 & x_{\text{Kamera},n,m} & y_{\text{Kamera},n,m} & 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \\ d \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} -x_{t,\text{Schlitten},1} \\ -y_{t,\text{Schlitten},1} \\ \dots \\ -x_{t,\text{Schlitten},1} \\ -y_{t,\text{Schlitten},1} \\ \dots \\ -x_{t,\text{Schlitten},2} \\ -y_{t,\text{Schlitten},2} \\ \dots \\ -x_{t,\text{Schlitten},m} \\ -y_{t,\text{Schlitten},m} \end{pmatrix}$$

Eine Lösung für dieses Gleichungssystem kann beispielsweise mit dem bereits in Kapitel 4.2.3 vorgestellten Levenberg-Marquardt Verfahren bestimmt werden.

Abschließend muss noch die Transformationsmatrix zur Überführung der Koordinaten aus dem Schlittensystem in das zweidimensionale Referenzsystem ermittelt werden. Die zu bestimmende Transformationsmatrix $T_{\text{Schlitten},\text{Ref}}$ lässt sich mittels der in Kapitel 4.2.3 vorgestellten Methodik zur Berechnung einer affinen Transformationsmatrix aus Punktentsprechungen im zweidimensionalen Vektorraum bestimmen. Dazu werden die Punktentsprechungen der bekannten Punkte der Referenzzyylinder im Referenzsystem und die im vorigen Schritt berechneten Koordinaten der Referenzpunkte im Schlittensystem genutzt.

Mit Kenntnis der Kameramatrix K , den Transformationsmatrizen $T_{\text{Kamera},\text{Schlitten}}$ und $T_{\text{Schlitten},\text{Ref}}$, der aktuellen Position des Schlittensystems $t_{\text{Schlitten}}$ sowie der Entfernung $z_0 \in \mathbb{R}^+$ der Referenzplattenoberfläche entlang der Z-Achse des Kamerasytems, kann nun mittels folgender Gleichung jeder beliebige Bildpunkt p_{Kamera} in einer Kameraaufnahme bei bekannter Entfernung zur Kamera z in Koordinaten des Referenzsystems überführt werden.

$$p_{\text{Ref}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & z - z_0 \end{pmatrix} \cdot T_{\text{Schlitten},\text{Ref}} \cdot \begin{pmatrix} 1 & 0 & x_{t,\text{Schlitten}} \\ 0 & 1 & y_{t,\text{Schlitten}} \\ 0 & 0 & 1 \end{pmatrix} \cdot T_{\text{Kamera},\text{Schlitten}} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot K^{-1} \cdot z \cdot p_{\text{Kamera}} \quad (4.14)$$

Die Entfernung z der Bildpunkte bezüglich des Kamerasytems ist für ein PCB in erster Näherung als konstant anzusehen, falls die Kamera senkrecht zur Oberfläche ausgerichtet ist. Die Entfernung muss also nur ein Mal zu Beginn der Analyse eines PCBs mittels der manuellen Messung des Abstands zweier Punkte und Selektion dieser im Kamerabild bestimmt werden (vgl. Gleichung (4.12)). Auch eine Berechnung von Bildkoordinaten aus Punkten des Referenzsystems ist möglich. Dazu muss Gleichung (4.14) entsprechend umformuliert werden:

$$p_{\text{Kamera}} = K \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & z_{\text{Ref}} + z_0 \end{pmatrix} \cdot T_{\text{Kamera}, \text{Schlitten}}^{-1} \cdot \begin{pmatrix} 1 & 0 & -x_{t, \text{chlitten}} \\ 0 & 1 & -y_{t, \text{Schlitten}} \\ 0 & 0 & 1 \end{pmatrix} \cdot T_{\text{Schlitten}, \text{Ref}}^{-1} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot p_{\text{Ref}} \quad (4.15)$$

Gleichung (4.15) liefert die Bildpunkte in homogenen Koordinaten der Form

$$p_{\text{Kamera}} = \begin{pmatrix} x_{\text{Kamera}} \\ y_{\text{Kamera}} \\ w \end{pmatrix}$$

Um die Pixelkoordinaten in der Bildebene zu erhalten, müssen diese abschließend noch normiert werden

$$p_{\text{Bild}} = \frac{1}{w} \cdot p_{\text{Kamera}} = \begin{pmatrix} \frac{x_{\text{Kamera}}}{w} \\ \frac{y_{\text{Kamera}}}{w} \\ 1 \end{pmatrix}$$

4.6. Evaluation

In diesem Kapitel werden die Ergebnisse der finalen Bewertung der aufgebauten Hardwareplattform in Kombination mit der entwickelten Kalibrierungsroutine vorgestellt. Zunächst werden in Kapitel 4.6.1 die eingesetzten Methoden zur Evaluation beschrieben. Dabei wurden sowohl theoretische, als auch praktische Verfahren genutzt. In Kapitel 4.6.2 werden anschließend die an der aufgebauten Kontaktierhardware gemessenen Ergebnisse vorgestellt. Das Verhalten des realen Prototypen wird dabei anhand der Messergebnisse unter Einbeziehung des Designentwurfs diskutiert.

4.6.1. Methodik

Die Kontaktierhardware wurde primär hinsichtlich der erzielten Positioniergenauigkeit und der Güte des Abgleichs zwischen dem Kamerasystem und den fahrbaren Prüfspitzen hin untersucht. Dazu wurden sowohl rein rechnerische, als auch praktische Testverfahren eingesetzt.

Theoretische Evaluation

Zur Beurteilung der entwickelten Kalibrierungsroutine in Zusammenhang mit der aufgebauten Kontaktierhardware erfolgte zunächst eine rein rechnerische Validierung der bestimmten Transformationsmatrizen. Für die kalibrierten Prüfeinheiten wurden dabei die durch elektrische Kontaktierung und anschließendes Kreis-Fitting bestimmten Mittelpunkte der vier mittleren Referenzzyliner mit den theoretischen, idealen Punkten auf der Referenzplatte verglichen. Dazu

wurden die gemessenen Punkte in das Referenzsystem mittels der bestimmten Transformationsmatrix $T_{\text{Prüfspitze},\text{Ref}}$ überführt und die Differenzen mit den idealen Referenzylinderkoordinaten gebildet.

Durch den Vergleich der Ergebnisse mehrfach durchgeföhrter Kalibrierungen können außerdem Aussagen über die Wiederholgenauigkeit der aufgebauten Hardwareplattform getroffen werden. Dazu wurden jeweils Testreihen von 10 Kalibrierungen pro Kontaktiereinheit durchgeföhr.

Das selbe Verfahren wurde ebenfalls mit dem fahrbaren Kamerasystem durchgeföhr. Dabei wird indirekt die Güte der Kombination aus den beiden bestimmten Transformationsmatrizen $T_{\text{Kamera},\text{Schlitten}}$ und $T_{\text{Schlitten},\text{Ref}}$ überprüft. Auf eine spezielle Ausrichtung der Kamera senkrecht zur Referenzplatte wurde an dieser Stelle bewusst verzichtet. Die Kameraausrichtung wird daher lediglich durch die konstruktiv eingearbeiteten Hilfsmittel zur Ausrichtung bei der Montage, wie etwa vorgesehene Anschlagkanten, bestimmt. Es lässt sich damit überprüfen, ob die Kontaktierhardware, auch von ungeschultem Personal zusammengebaut, zuverlässig funktioniert. Dies ist insbesondere im Hinblick auf die Veröffentlichung als Open Hardware-Projekt interessant.

Praktische Evaluation

Zusätzlich zu den theoretisch ermittelten Genauigkeitswerten wurden praktische Tests durchgeföhr. Hierfür wurde von einem Mitarbeiter der *SCHUTZWERTH GmbH* eine spezielle Testplatine entworfen, die in Abbildung 4.8 zu sehen ist. Auf dieser Platine befinden sich zwei unbestückte BGA- und insgesamt vier SMD-Footprints. Die zugehörigen Pads sind mit einem Durchmesser von 0,25 mm bzw. einer Größe von $0,2 \times 0,53$ mm repräsentativ für die zu kontaktierenden Chip Packages. Die Pads sind mit insgesamt vier, in Reihe geschalteten Schieberegistern verbunden, welche über die in Form eines Steckers nach außen geföhrten Kontakte ausgelesen werden können.

$4 \times \text{SMD}$ (je 8 Pads mit $0,2 \times 0,53$ mm, Rasterabstand $x = 0,67$ mm, $y = 0,35$ mm)

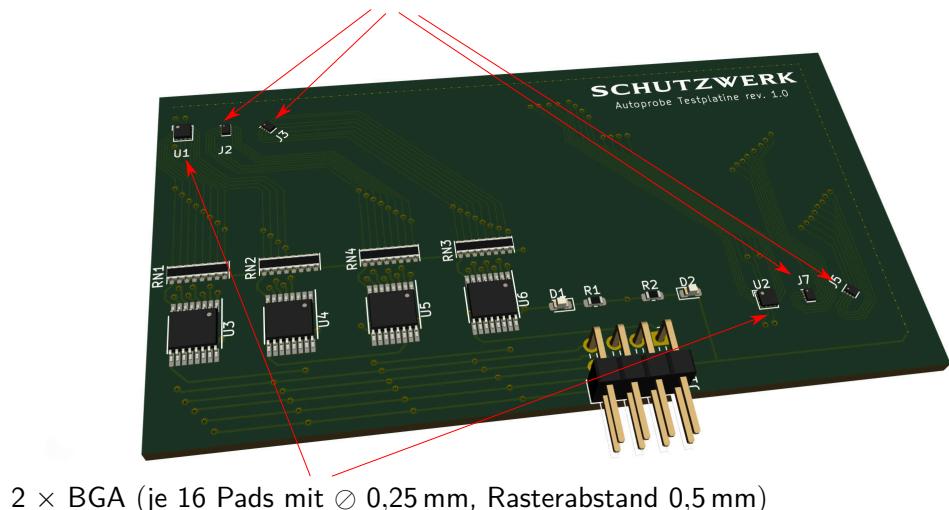


Abbildung 4.8.: Rendering der entworfenen Testplatine

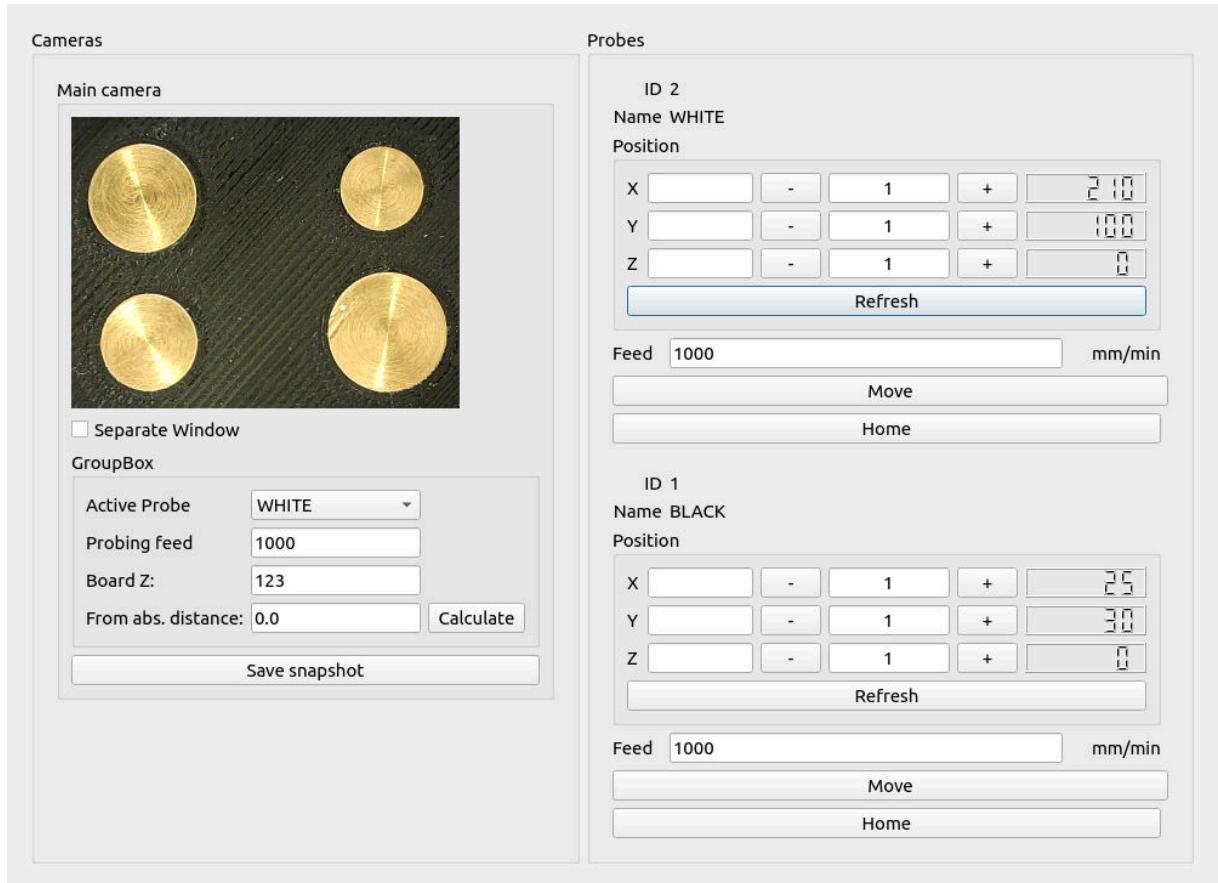


Abbildung 4.9.: Screenshot der entwickelten Software zur interaktiven Steuerung der Kontaktierhardware

Zur Überprüfung der Genauigkeit der durchgeföhrten Kalibrierung wurde eine Software mit grafischer Oberfläche entwickelt. Über die UI ist es möglich, die beiden Prüfspitzen manuell zu verfahren. Weiterhin kann das Bild des eingesetzten Kamerasytems live verfolgt werden. Für Tests zur Positioniergenauigkeit und den Abgleich der fahrbaren Einheiten mit dem Kamerasytem wurde außerdem eine spezielle Routine implementiert. Durch einen Mausklick auf einen beliebigen Punkt des Live-Kamerabilds werden die Zielkoordinaten für eine zuvor ausgewählte Prüfspitze berechnet. Anschließend wird diese automatisch an die bestimmten Zielkoordinaten bewegt. Die UI der entwickelten Software ist in Abbildung 4.9 zu sehen.

Die Pads auf der beschriebenen Testplatine sind sehr klein. Durch eine reine Sichtprüfung ist es nur schwer zu erkennen, ob ein zuvor manuell ausgewähltes Testpad erfolgreich mit einer Prüfspitze kontaktiert wurde. Der Zustand der Testplatine wird daher nach jeder ausgeführten, interaktiven Kontaktierung ausgegeben.

Mittels der beschriebenen interaktiven Kontaktierung wurden Tests der Positioniergenauigkeit durchgeführt. Dabei wurde die Kameraeinheit zunächst so verfahren, dass die zu testenden Pads der Testplatine im Kamerabild sichtbar waren. Anschließend wurden die Testpads manuell selektiert und durch das System automatisch kontaktiert. Nach jeder Selektion wurde der Zustand der Testplatine auf eine erfolgreiche Kontaktierung hin überprüft. Dieser Test wurde

mit beiden Kontaktiereinheiten durchgeführt.

4.6.2. Ergebnisse

Die Ergebnisse der durchgeführten Evaluation der aufgebauten Hardwareplattform in Kombination mit der entwickelten Kalibrierungsroutine sind in diesem Kapitel zusammengefasst. Zunächst wird auf die im Rahmen der theoretischen Evaluation bestimmten Positioniergenauigkeiten der einzelnen Einheiten eingegangen. Anschließend werden die Ergebnisse der interaktiv durchgeführten, praktischen Tests vorgestellt.

Ergebnisse der theoretischen Evaluation

Die Ergebnisse der theoretischen Evaluation von 10 nacheinander ausgeführten Kalibrierungen für die beiden Kontaktiereinheiten und das fahrbare Kamerasytem sind in Abbildung 4.10 dargestellt. In den Diagrammen ist der Fehler zwischen den idealen und den mittels den Messpunkten der Kalibrierung und den bestimmten Transformationsmatrizen berechneten theoretischen Pinzentren in der XY-Ebene eingetragen. Die Abweichungen sind für jeden Referenzpin separat visualisiert, um eventuelle Trends oder einseitige Verschiebungen besser zu erkennen. Die Nummerierung der Pins entspricht denen auf der Referenzplatte (vgl. Abbildung 4.2). Die Achsangaben beziehen sich auf das Koordinatensystem der Hardwareplattform.

Zusätzlich sind in Tabelle 4.1 die über alle Referenzpins bestimmten, maximalen Spannweiten der Fehler in der X- und Y-Achse aufgelistet. Für die Kontaktiereinheiten ist auch die Spannweite der Fehler in der Z-Richtung bestimmt worden. Die Spannweite wird dabei als Differenz zwischen dem größten und dem kleinsten gemessenen Fehler in den Achsen des Referenzkoordinatensystems bestimmt. Da die Höhe der Kamera bei der Kalibrierung nur einmalig bestimmt wird, konnte für diese der Fehler in der Z-Achse nicht bestimmt werden. Ungenauigkeiten in der Kamerahöhe wirken sich indirekt durch die projektive Geometrie auf Fehler in den X- und Y-Koordinaten aus.

Einheit	max. Fehlerspannweite in X in mm	max. Fehlerspannweite in Y in mm	max. Fehlerspannweite in Z in mm
Kamera	0,022	0,180	-
Einheit <i>WEISS</i>	0,091	0,091	0,288
Einheit <i>SCHWARZ</i>	0,057	0,053	0,389

Tabelle 4.1.: Maximale Fehlerspannweiten der aus Messpunkten berechneten Referenzpinkoordinaten

Die Fehlerspannweiten der beiden Kontaktiereinheiten sind mit Werten von 0,09 bzw. max. 0,57 mm für die Kontaktiereinheit *WEISS* und *SCHWARZ* in der XY-Ebene gering und unterschreiten die geforderte Positioniergenauigkeit von 0,1 mm. Abweichungen in der Z-Achse

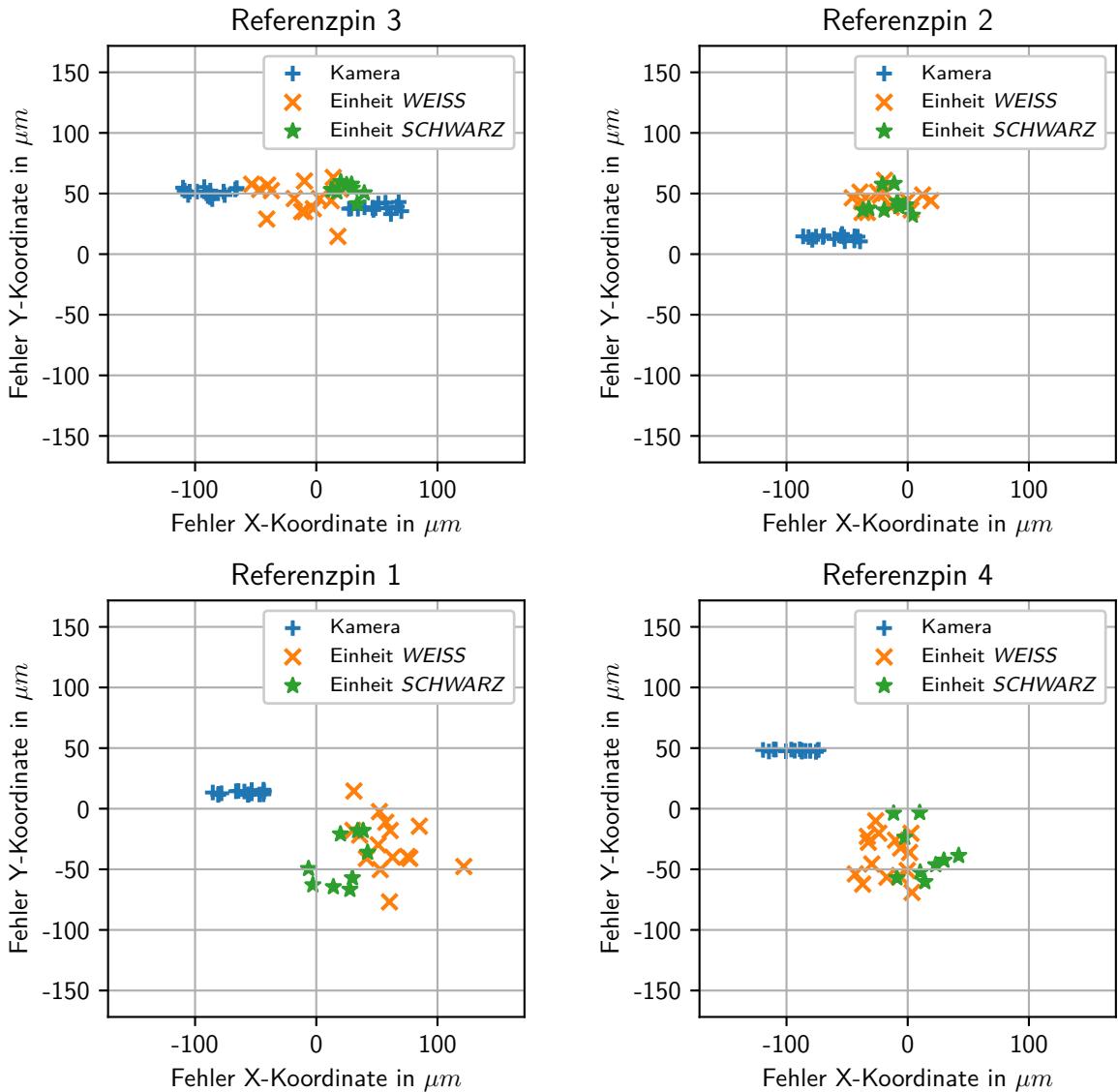


Abbildung 4.10.: Fehler der aus Messpunkten berechneten Referenzpintoordinaten

spielen für die spätere Kontaktierung nur eine untergeordnete Rolle. Durch den Einsatz von gefederten Kontaktierstiften können Höhendifferenzen von bis zu 2 mm problemlos ausgeglichen werden. Die Kontaktiereinheiten zeigten beide in der Z-Achse mit 0,288 bzw. 0,389 mm vergleichsweise große Fehler.

In der Y-Achse zeigt sich in den Messwerten der Kontaktiereinheiten für die Referenzpins 2 und 3 ein positiver bzw. bei den Referenzpunkten 1 und 4 ein negativer Versatz. Die Ursache hierfür ist vermutlich ein Fehler in der 3D-gedruckten Referenzplatte. Entweder wurde sie von dem eingesetzten 3D-Drucker nicht präzise genug gefertigt oder es kam zu Verzug während des Abkühlvorgangs der Kunststoffplatte.

Insbesondere bei dem größten Referenzpin 1 mit einem Durchmesser von 10 mm streuen die Messwerte der Kontaktiereinheiten stärker. Dies deutet auf Schwierigkeiten bei der Bestimmung des Pinzentrums durch das eingesetzte Kreis-Fitting. Ein weiterer Grund könnte die unsaubere

Platzierung des Referenzzyllinders in der Referenzplatte sein. Ist der Pin etwa leicht schräg montiert, kann es zu einem Abrutschen der Prüfnadel bei der binären Suche der Kantenpunkte kommen. Eine senkrecht angeordnete Prüfspitze oder eine nach der Montage der Referenzpins plangeschliffene Referenzplatte könnte diese Fehlerquellen eliminieren.

Die Messwerte der Kontaktiereinheit *WEISS* streuen stärker als die der Einheit *SCHWARZ*. Beide Einheiten sind bezüglich ihrer mechanischen Konstruktion identisch. In der aufgebauten Plattform wurden jedoch unterschiedliche Motoren in den beiden Achsen eingebaut. Die Einheit *WEISS* wird mit NEMA 14 Schrittmotoren betrieben. Das Achssystem der Kontaktiereinheit *SCHWARZ* wird von NEMA 17 Motoren angetrieben. Eine Ursache für die unterschiedlich starken Streuungen der Messwerte könnten daher in den mechanischen Toleranzen der Motoren liegen, die je nach Motorgröße und Hersteller variieren.

Auch für das kalibrierte Kamerasytem wurden die aus den ermittelten Transformationen berechneten Referenzpunktkoordinaten mit den idealen Werten zum Zwecke der Evaluation verglichen. In Y-Richtung konnte eine mit 0,022 mm sehr geringe Fehlerspannweite erzielt werden. Der Wert für die maximale Fehlerspannweite von 0,18 mm in der X-Achse ist jedoch relativ groß. Grund hierfür sind die Messungen bei Referenzzyylinder 3 (vgl. Abbildung 4.10). Die Messpunkte bilden dort zwei separate Punktwolken. Dieses Phänomen tritt nur an diesem Referenzpunkt auf. Bei der Inspektion der Referenzplatte konnten keine Besonderheiten festgestellt werden. Es ist zu vermuten, dass numerische Ungenauigkeiten in den Transformationsmatrizen die Ursache hierfür sind. Diese Vermutung konnte jedoch bisher nicht verifiziert werden.

Die Entfernungsbestimmung der Referenzplatte in Z-Richtung geht gemäß Gleichung (4.14) indirekt in die Genauigkeit der X- und Y-Achse ein. Da die Abweichungen in diesen Achsen, mit Ausnahme der Messungen des 3. Referenzpunktes, nur sehr gering sind, lässt dies auch auf eine sehr gute Bestimmung der Entfernung in Z-Richtung schließen.

In diesem Abschnitt wurden lediglich die Positionier- und Wiederholgenauigkeiten bezogen auf das künstliche geschaffene Referenzsystem betrachtet. Für den praktischen Einsatz der Kontaktierhardware ist eine hohe Genauigkeit der gesamten Transformation von Koordinaten aus dem Kamerabild in das System einer Prüfspitze essentiell. Um diese bewerten zu können, wurden zusätzlich praktische Tests für diesen Anwendungsfall durchgeführt.

Ergebnisse der praktischen Evaluation

Es konnten alle Testpads auf der entworfenen Testplatine erfolgreich mit beiden Prüfeinheiten nach manueller Selektion der Testpadzentren im Kamerabild kontaktiert werden. Der Abgleich des Kamerasytems mit den Koordinatensystemen der fahrbaren Kontaktiereinheiten ist daher mit einer Genauigkeit von weniger als $\pm 0,1$ mm durch die entwickelte Kalibrierungsroutine möglich. Andernfalls hätten die Testpads mit einer minimalen Größe von 0,2 mm nicht kontaktiert werden können. Die Ergebnisse der praktischen Evaluation bestätigen damit die bereits bei der theoretischen Evaluation ermittelten Genauigkeitswerte von maximal $\pm 0,1$ mm des Kamerasytems.

4.7. Image Stitching zur Generierung hochauflöster PCB-Aufnahmen

Durch Wahl eines bewegten Kamerasytems mit einem recht geringen FOV in dem aufgebauten Prototypen ist es für eine Analyse eines kompletten PCBs notwendig, mehrere Kameraaufnahmen zu einem Gesamtbild zusammenzusetzen. Dieser Vorgang wird auch als *Image Stitching* bezeichnet. Für diesen Anwendungsfall wurden bereits zahlreiche Verfahren entwickelt [48, S. 375-406]. Die Methoden erhalten als Eingangsdaten lediglich einen Satz von Bildaufnahmen ohne weitere Metadaten. Je nach Verfahren werden dann unterschiedliche Bewegungsmodelle genutzt, um die Kameraposen der Bildaufnahmen zu rekonstruieren und ein Gesamtbild zu konstruieren.

Durch die in Kapitel 4.5 im Zuge der Kalibrierung bestimmten Transformationen ist für jede Bildaufnahme des Kamerasytems die Kamerapose bekannt. Basierend auf diesem Metawissen wird in diesem Kapitel ein sehr einfaches Verfahren für die Generierung hochauflöster Bildaufnahmen eines PCBs mit der entwickelten Hardwareplattform vorgestellt.

4.7.1. Eingesetztes Verfahren

In Abbildung 4.11 ist der Aufbau und Ablauf des entwickelten Verfahrens für das *Image Stitching* dargestellt. Zunächst wird der zu scannende Bereich in der XY-Ebene des Referenzsystems definiert. Weiterhin wird die gewünschte Auflösung des finalen Bildes in mm pro Pixeln vorgegeben. Daraufhin wird ein leeres Bild mit der aus den Abmessungen des Scanbereichs und der gewählten Scanauflösung berechneten Größe in Pixeln im Speicher angelegt. Für jedes Pixel des finalen Bildes können die entsprechenden Koordinaten im Referenzsystem berechnet werden.

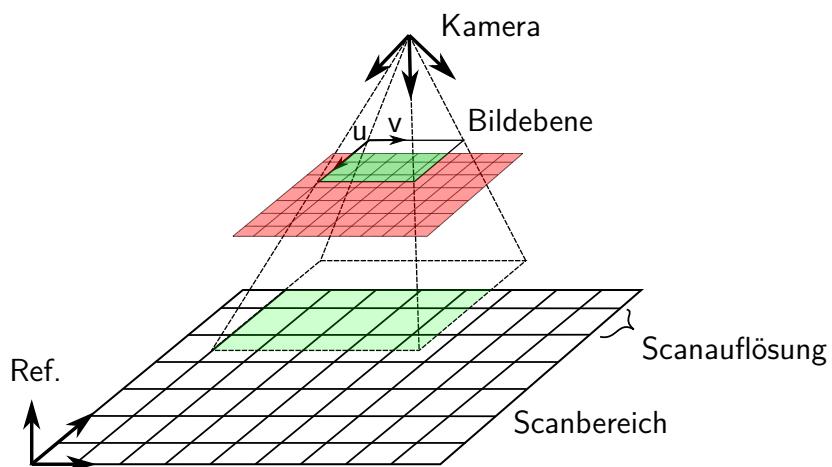


Abbildung 4.11.: Schematischer Ablauf des Image Stitching Verfahrens

Die fahrbare Kamera rastert anschließend den definierten Scanbereich mit einer fixen, vor-konfigurierten Schrittweite ab. An jeder Position wird dabei eine Bildaufnahme erstellt. Mithilfe

von Gleichung (4.15) werden die Koordinaten der Pixel des finalen Bildes in Form der entsprechenden Koordinaten des Referenzsystems in das jeweilige Kamerabild transformiert. Dabei entstehen Koordinaten in der Bildebene mit Subpixelgenauigkeit. Für jedes Pixel des finalen Bildes, das nach der Transformation im aktuellen Kamerabild sichtbar ist (vgl. Abbildung 4.11, grüner Bereich der Bildebene), wird der Farbwert mittels der in Kapitel 4.2.4 vorgestellten bilinearen Interpolation aus den Farbwerten der umgebenden Pixel des Kamerabildes berechnet. Der interpolierte Farbwert wird abschließend dem Pixel im finalen Bild zugewiesen. Bereiche im aktuellen Kamerabild, die nicht im Scancbereich liegen, wie in Abbildung 4.11 rot dargestellt, werden ignoriert.

Das Verfahren macht es möglich, eine sehr hoch aufgelöste Aufnahme des zu untersuchenden PCBs zu erstellen. Der visuell erfassbare Bereich wird dabei lediglich durch den Verfahrweg des Kameraschlittens begrenzt. Die Auflösung bleibt unabhängig von der Größe des untersuchten PCBs und kann vor Beginn des Scancorgangs in der absoluten Einheit Pixel pro mm entsprechend den Anforderungen gewählt werden. Auch mit einer relativ niedrig auflösenden Kamera kann damit eine hohe, reale Auflösung in Pixeln pro mm erreicht werden. Die Kamera wird dazu so montiert bzw. mit einem entsprechenden Objektiv ausgestattet, dass durch das FOV und die Kameraauflösung die geforderte maximale reale Auflösung erzielt wird. Das FOV wird dementsprechend bei einem gering auflösenden Kamerasystem ebenfalls recht klein sein. Die Anzahl der Einzelaufnahmen und die Zeit für den gesamten Scancorgang steigt damit. Die Optik des Kamerasystems muss nicht, wie bei Zoomobjektiven, auf jede PCB-Größe neu eingestellt werden. Somit genügt eine initiale Bestimmung der Kameramatrix K , was den Kalibrierungsaufwand gering hält.

Bei der Zusammenführung mehrerer Kameraaufnahmen entstehen sehr häufig Artefakte, die bei einer einzelnen Bilddurchsuche nicht zu finden sind. Fehler in der Transformation der Koordinaten des Referenzsystems in die Bildebene der Kamera führen zu Stufen an den Übergängen der zusammengesetzten Bilder. Auch Unterschiede in den Lichtverhältnissen der Einzelaufnahmen führen dazu, dass die Grenzen der einzelnen Aufnahmen in dem zusammengeführten Bild deutlich erkennbar sind. Da in Gleichung (4.15) die Transformation aus dem Referenzsystem in die Bildebene als Matrixmultiplikation formuliert ist, lässt sich diese auch für eine große Anzahl an Punkten sehr effizient auf aktueller Hardware durchführen. Die anschließende bilineare Interpolation der Pixelwerte ist jedoch für viele Pixel vergleichsweise aufwendig, was den Scancorgang merklich verzögert.

4.7.2. Evaluation

Für die Evaluation des *Image Stitching*-Verfahrens wurde eine weitere Testplatine genutzt. Diese wird zusammen mit der Beschreibung des Evaluationsablaufs im nachfolgenden Abschnitt zuerst beschrieben. Die Vorstellung der Ergebnisse anhand einer mit dem Verfahren erzeugten Bilddurchsuche erfolgt im zweiten Teil dieses Kapitels.

Methodik

Bei der Evaluation des *Image Stitching*-Verfahrens kam eine Testplatine zum Einsatz, welche in Abbildung 4.12 zu sehen ist. Es handelt sich dabei um eine unbestückte Breakout-Platine für TQFP Chips mit einem Rasterabstand von 0,5 mm und einer Padbreite von 0,2 mm. Die Leiterbahnen und länglich ausgeführten Pads sind in einer regelmäßigen Struktur angeordnet und zeichnen sich sehr gut auf dem PCB ab. Es wurde außerdem eine zweifarbige LED an der Platine angelötet. Die Verschaltung ist dabei so gewählt, dass bei Kontaktierung der Testpads entlang des Padrasters in horizontaler oder vertikaler Richtung die LEDs jeweils abwechselnd aufleuchten.

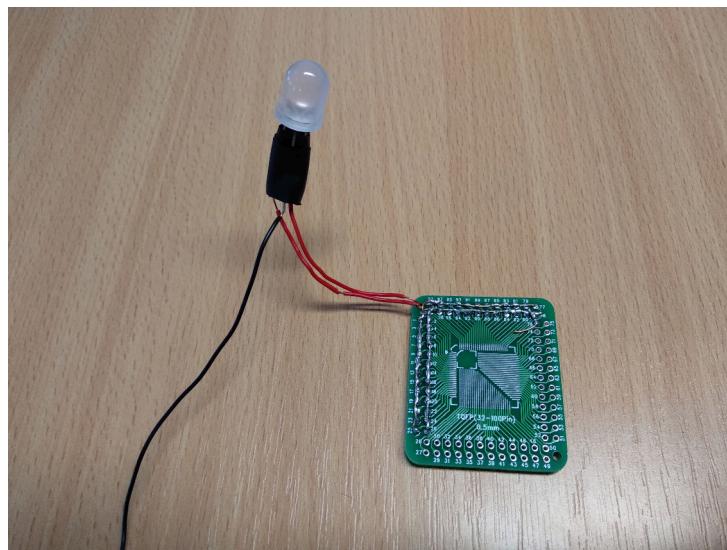


Abbildung 4.12.: Genutzte Platine zur Evaluation des *Image Stitching*-Verfahrens

Im Zuge der Evaluation wurde zunächst ein mit dem beschriebenen *Image Stitching*-Verfahren generiertes Bild aufgenommen. Anschließend erfolgte zunächst eine visuelle Überprüfung hin bezüglich auftretender Bildartefakte an den Übergängen zwischen den einzelnen Aufnahmen. Anschließend wurde die in Kapitel 4.6.1 vorgestellte Software für die interaktive elektrische Kontaktierung mit der Hardwareplattform genutzt. Dabei wurden anstelle von Punkten im Live-Kamerabild Punkte in dem generierten Bild ausgewählt. Es wurden jeweils fünf direkt nebeneinander liegende Pads sowohl im horizontalen, als auch vertikalen Padraster für die Kontaktierung ausgewählt.

Die Bildkoordinaten des künstlich generierten Gesamtbildes sind dabei bis auf einen Skalierungsfaktor und einen Versatz in der XY-Ebene identisch mit den Koordinaten des Referenzsystems. Der Skalierungsfaktor wird durch die gewählte Scanauflösung festgelegt. Der Versatz in der XY-Ebene ergibt sich durch die Größe und Position des gewählten Scanbereichs. Eine Berechnung der Positionen für die Prüfspitzen konnte daher direkt mit Gleichung (4.8) durchgeführt werden. Ob ein Pad getroffen wurde, konnte durch die LED der Testplatine überprüft werden.

Ergebnisse

Ein mit dem beschriebenen *Image Stitching*-Verfahren erzeugtes, zusammengesetztes Bild der Breakout-Platine zeigt Abbildung 4.13.

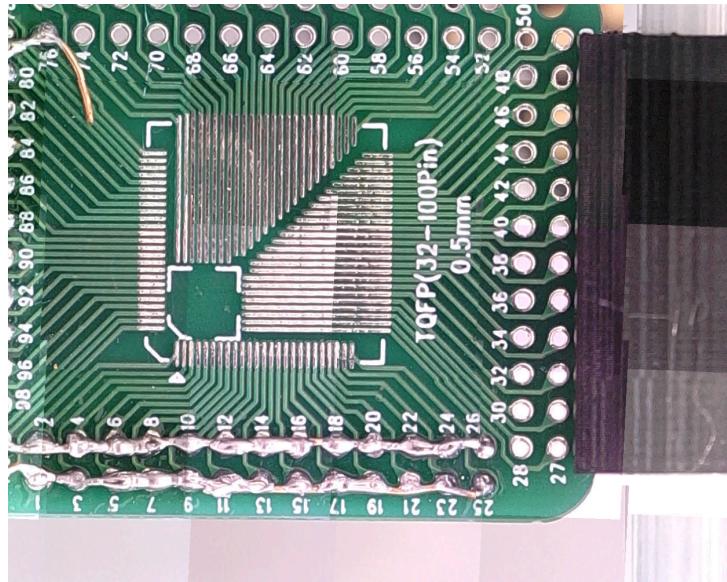


Abbildung 4.13.: Zusammengefügtes Bild aus mehreren Einzelaufnahmen des fahrbaren Kamerasystems der Kontaktierplattform

Bis auf einige Artefakte am Bildrand liefert der Algorithmus ein sehr gutes Bild. An den gut sichtbaren Leiterbahnen der Platine sind zwischen den einzelnen Aufnahmen nur minimale Stufen erkennbar, was auf eine gute Kalibrierung des fahrbaren Kamerasystems schließen lässt. Die sichtbaren Artefakte am Bildrand könnten durch nicht berücksichtigte Linsenverzerrungen entstanden sein. Ein generelles Problem beim *Image Stitching* sind variierende Lichtverhältnisse. Auch Abbildung 4.13 zeigt aufgrund unterschiedlicher Beleuchtungsverhältnisse sichtbare Helligkeitsunterschiede zwischen den Einzelaufnahmen.

Bei der interaktiven Selektion von Punkten in dem zusammengeführten Bild und anschließenden automatischen Kontaktierung wurden alle ausgewählten Pads erfolgreich kontaktiert. Der Abgleich zwischen den Koordinaten des generierten Bildes und den Kontaktiereinheiten muss sich damit in einem Bereich von weniger als $\pm 0,1$ mm befinden. Ein erfolgreiche Kontaktierung der Pads mit einer Breite von 0,2 mm wäre sonst nicht möglich gewesen.

4.8. Zusammenfassung und Ausblick

Die Kalibrierung ist zentraler Bestandteil der entwickelten Hardwareplattform für die automatisierte IT-Sicherheitsanalyse. In diesem Kapitel werden nochmals die mit dem Gesamtsystem, bestehend aus Hardwareplattform und Kalibrierung, erzielten Ergebnisse zusammengefasst. Weiterhin werden Anregungen für Verbesserungen der Kalibrierungsroutine und des vorgestellten Verfahrens für das *Image Stitching* gegeben.

Kalibrierungsroutine

Mit der entwickelten Kalibrierungsroutine in Kombination mit dem aufgebauten Hardwareprototypen konnten sehr gute Ergebnisse bezüglich der Positioniergenauigkeit erzielt werden. Der Abgleich des Kamerakoordinatensystems zu den Systemen der Prüfspitzen war mit einer Genauigkeit von weniger als $\pm 0,1$ mm in der XY-Ebene möglich.

Die durchgeführten Evaluationen fanden alle in einem sehr kleinen Verfahrbereich statt. Dies ist für eine erfolgreiche automatisierte Kontaktierung von kleinen PCBs bereits ausreichend. Durch Ausweitung der Messpunkte und großflächigere Verteilung im Arbeitsraum könnten geometrische Fehler noch besser erfasst werden.

Mit der entwickelten Kalibrierungsroutine können derzeit nur statische Fehler, wie etwa Ungenauigkeiten in der Rechtwinkligkeit der montierten Achsen, ausgeglichen werden. Um eine umfassendere Aussage über die Genauigkeit der entwickelten Hardwareplattform treffen zu können, müssen weitere Tests durchgeführt werden. Beispielsweise sollte das Verhalten der Hardware bezüglich schlechender, sich aufkumulierender Fehler durch Schrittverluste bei längerem Betrieb untersucht werden.

Bei der Kalibrierung des Kamerasystems wird angenommen, dass sich dieses orthogonal zur Referenzplatte befindet. Eine Posenbestimmung der Kamera mittels der vier Referenzzyliner könnte Aufschluss über die relative Orientierung der Kamera liefern. Mit diesen Informationen könnte die Kameraausrichtung nachträglich manuell korrigiert werden. Negative Einflüsse auf die restliche Kalibrierung werden dadurch minimiert.

Image Stitching

Die im Rahmen der Evaluation des *Image Stitching*-Verfahrens generierten Bildaufnahmen zeigten eine gute Qualität bezüglich der Positionierung der Einzelaufnahmen in dem Gesamtbild. Der Versatz zwischen den Einzelaufnahmen war am Rand des Scancbereichs jedoch deutlich größer als in der Mitte. Die Fehlerursache hierfür muss weiter untersucht werden. Der Abgleich zwischen den Koordinatensystemen und dem künstlich durch das Verfahren generierten Bildes ist ebenfalls gut. Bei interaktiven Tests auf einer Breakout-Platine konnten Pads mit einer Größe von 0,2 mm sicher kontaktiert werden.

Um durch Linsenverzeichnungen verursachte störende Effekte zu minimieren, könnten die bisher nicht in dem Verfahren berücksichtigten nichtlinearen Kameraparameter in die Bestimmung der Transformation zwischen Punkten des Referenzsystems und der Bildebene einbezogen werden. Dies würde jedoch zu einem gesteigerten Rechenaufwand führen. Alternativ könnte auch nur ein relativ kleiner Bildausschnitt um den Bildhauptpunkt herum für die finale Bildgenerierung verwendet werden, da in diesem Bereich der Einfluss durch Linsenverzerrung geringer ist.

Zum Ausgleich von unterschiedlichen Lichtverhältnissen zwischen den Aufnahmen könnten die Schritte, mit denen der Scancbereich abgerastert wird, entsprechend kleiner gewählt werden. Damit liegen für jedes Pixel im finalen Bild mehrere Farbwerte vor. Durch Bildung des

Mittelwertes könnten Schwankungen in der Beleuchtung ausgeglichen und ein gleichmäßigeres Ausgabebild erzielt werden. Auch ein nachträglich durchgeführter Beleuchtungsabgleich ist an dieser Stelle denkbar, um die Bildqualität weiter zu verbessern.

Die Geschwindigkeit des Verfahrens wird zu einem großen Teil durch die für die Zusammenführung der Einzelaufnahmen notwendigen Rechenoperationen bestimmt. Eine Möglichkeit zur Steigerung der Geschwindigkeit des Verfahrens könnte die Auslagerung dieser Operationen auf die *Graphics Processing Unit* (GPU) sein, da diese speziell für Matrixmultiplikationen optimiert ist. Auch die Aufgabenstellung der Interpolation eines Farbwertes aus einem Bild findet sich in der Computergrafik bei der Texturierung von Objekten wieder. GPUs können daher auch für diesen Anwendungszweck effizient genutzt werden.

Mit dem entwickelten Verfahren konnten in einem kleinen Scancbereich gute Ergebnisse erzielt werden. Für eine finale Bewertung des eingesetzten *Image Stitching*-Verfahrens als Methode zur Bildgenerierung im Kontext der automatisierten Sicherheitsanalyse müssen jedoch weitere Tests durchgeführt werden. Es ist insbesondere zu untersuchen, ob und in wie weit die Bildartefakte und abrupten Helligkeitssprünge in dem erzeugten Bild andere Verfahren, wie etwa die automatische Chiperkennung mit Neuronalen Netzen, beeinflusst.

5. Zusammenfassung

Mit dieser Arbeit konnte gezeigt werden, dass sich sowohl die visuelle, als auch elektrische Analyse hinsichtlich der IT-Sicherheit von PCBs in großen Teilen automatisieren lässt. Bei der Erkennung einzelner Komponenten auf vollbestückten PCBs konnten mit technischen Neuronalen Netzen sehr gute Ergebnisse erzielt werden. Die zusammengestellten Bilddatensätze erwiesen sich im direkten Vergleich mit einem bereits bestehenden PCB-Datensatz als besserer Trainingsinput für die untersuchten Modelle. Durch die Evaluation auf einem zeitgemäßen und praxisnahen Datensatz konnte der Nutzen für den Einsatz in der täglichen Praxis eines Sicherheitsanalysten bewiesen werden. Es konnte weiterhin gezeigt werden, dass dieses Verfahren klassischen, Computer Vision basierten Methoden überlegen ist. So zeigten die untersuchten Neuronalen Netze eine deutlich erhöhte Robustheit gegenüber bekannten Störquellen in PCB-Bildaufnahmen, wie etwa unterschiedliche Lötstopfarben oder schwankenden Lichtverhältnissen.

Für die Aufgabenstellung der Lokalisierung einzelner Komponentenpins wurden unterschiedliche Verfahren evaluiert. Mit einer entwickelten Computer Vision Pipeline und zwei trainierten Neuronalen Netzen wurden Verfahren vorgestellt, mit denen diese zeitaufwendige Tätigkeit größtenteils automatisiert werden kann. Die vorgestellten Verfahren müssen in zukünftigen Arbeiten weiter verbessert werden. So muss die Erkennungsrate erhöht und insbesondere die Robustheit der Verfahren bei unterschiedlichen Chip Packages gesteigert werden. Der erstellte Bilddatensatz und die genutzten Evaluationsmetriken können dabei als Ausgangspunkt genutzt werden.

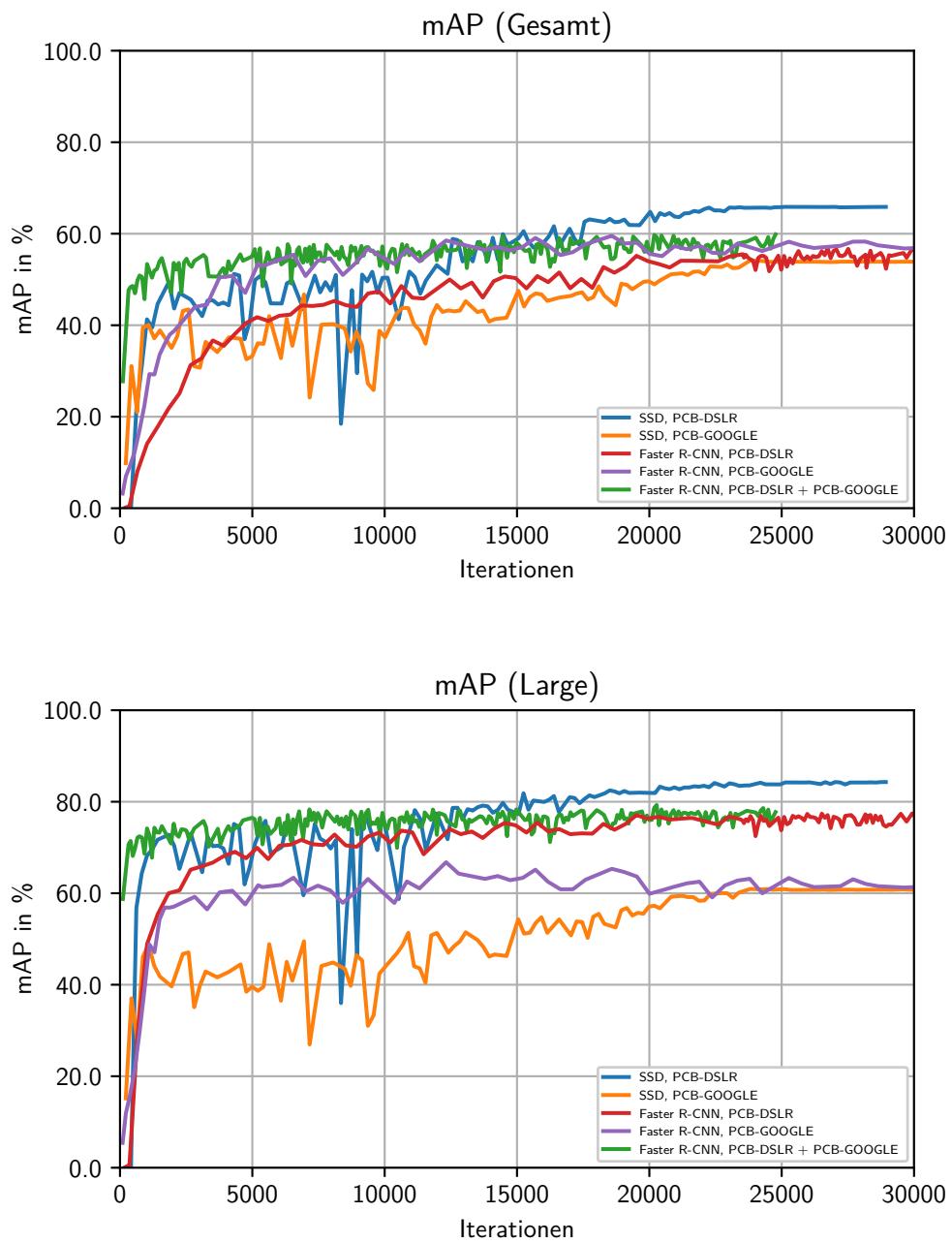
Basierend auf den gewonnenen Erkenntnissen bei der automatisierten, visuellen Analyse von bestückten PCBs wurden Anregungen für mögliche Weiterentwicklungen auf diesem Gebiet gegeben. Neuronale Netze bieten hierbei großes Potential hinsichtlich einer Steigerung des Automatisierungsgrades der PCB-Analyse oder des *Reverse Engineering*. Dazu werden mehr Trainings- und auch Evaluationsdaten benötigt. Durch eine speziell für diese Aufgabe angepassten Netztopologie könnte auch die Analysedauer reduziert werden.

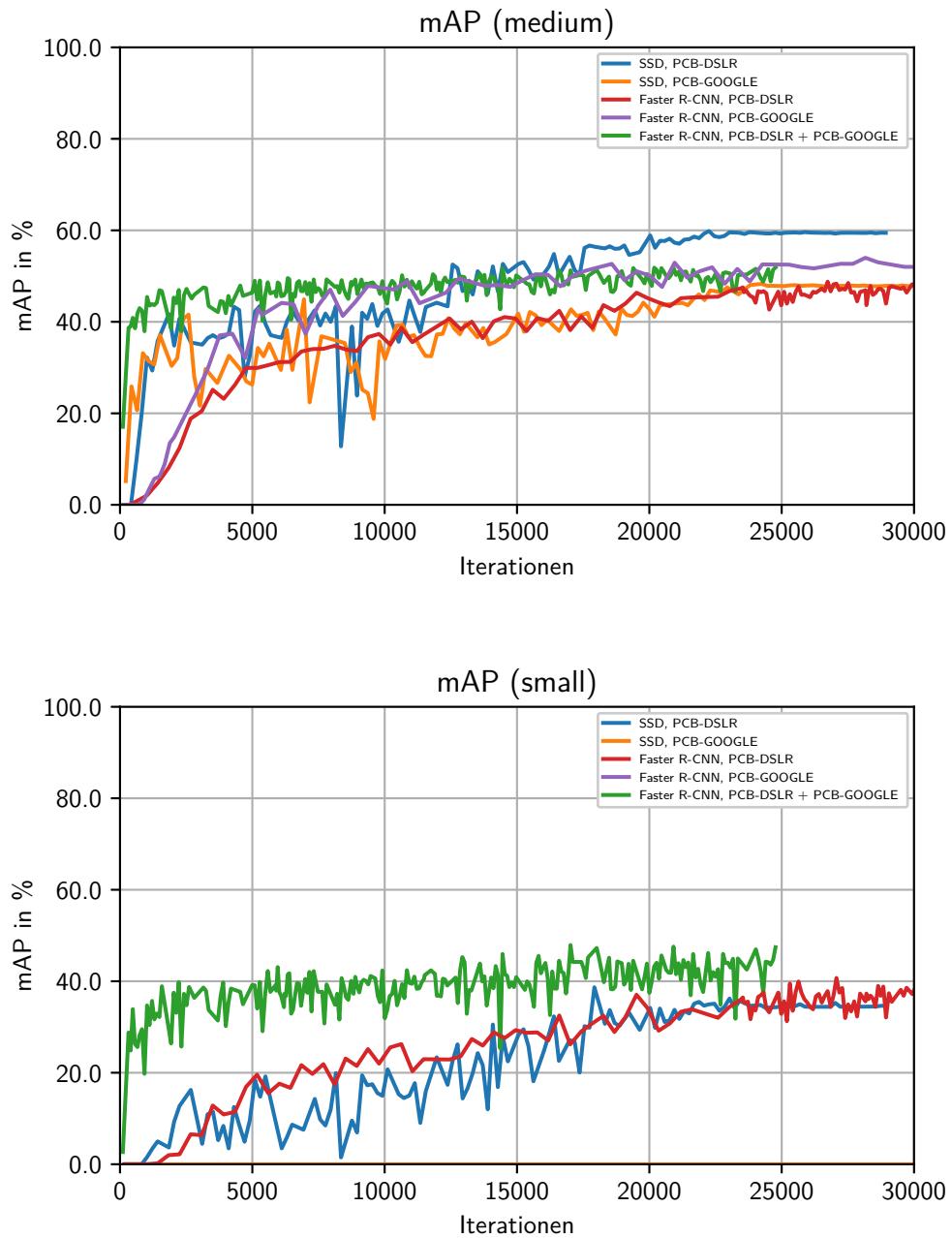
Mit der entwickelten, modularen Hardwareplattform wurde der Grundstein für automatisierte, elektronische und elektrische Tests von PCBs gelegt. Der einfache Aufbau in Kombination mit der entwickelten Kalibrierungsroutine erlaubt eine sehr einfache Replikation des Systems ohne besondere Kenntnisse. Dies fördert weitere Forschungen und Entwicklungen basierend auf dieser Plattform. Mit der entwickelten Software zur teilautomatisierten Analyse waren interaktive Kontaktierungen von Testplatinen mit zuverlässiger Trefferquote möglich. Die mit der Kalibrierungsroutine erreichte Genauigkeit des Abgleichs der Koordinatensystem der einzelnen

Einheiten war sehr gut. Es konnten erfolgreich Pads mit einer Größe von 0,2 mm durch das System kontaktiert werden. Um den Nutzen der Kontaktierhardware für den praktischen Einsatz zu steigern, müssen weitere Analysesysteme, wie etwa der JTAGulator¹, integriert werden. Weiterhin wurden Ideen für mechanische Erweiterungen genannt, die das Einsatzspektrum des Systems bei der IT-Sicherheitsanalyse noch weiter vergrößern können.

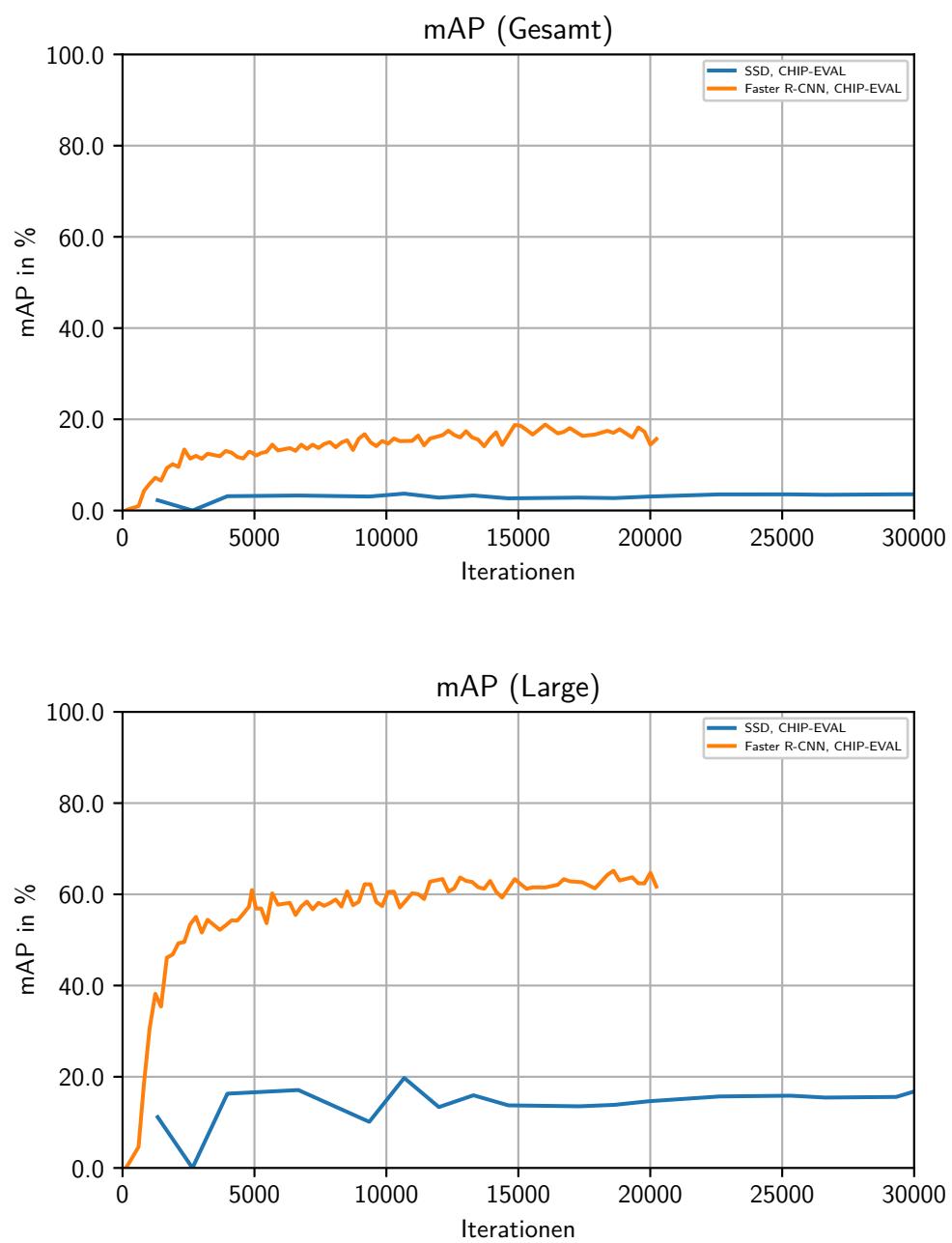
¹<http://www.jtagulator.com/>

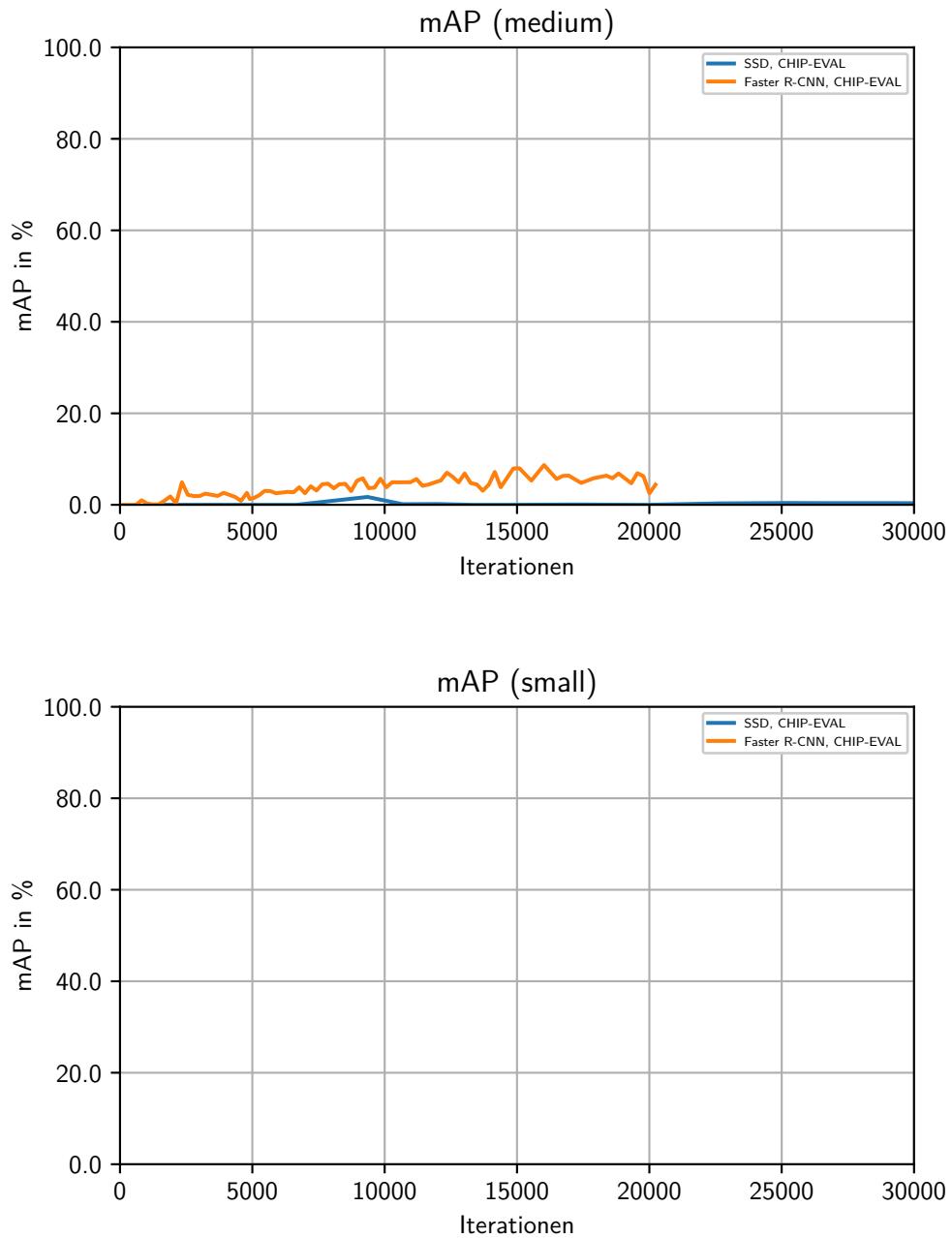
A. Trainingsverlauf der Neuronalen Netze zur Chiperkennung





B. Trainingsverlauf der Neuronalen Netze zur Pinerkennung





C. Motorkennline Schrittmotor Nanotec - ST3518M1004-A

ST3518M1004-A - Schrittmotor – NEMA 14

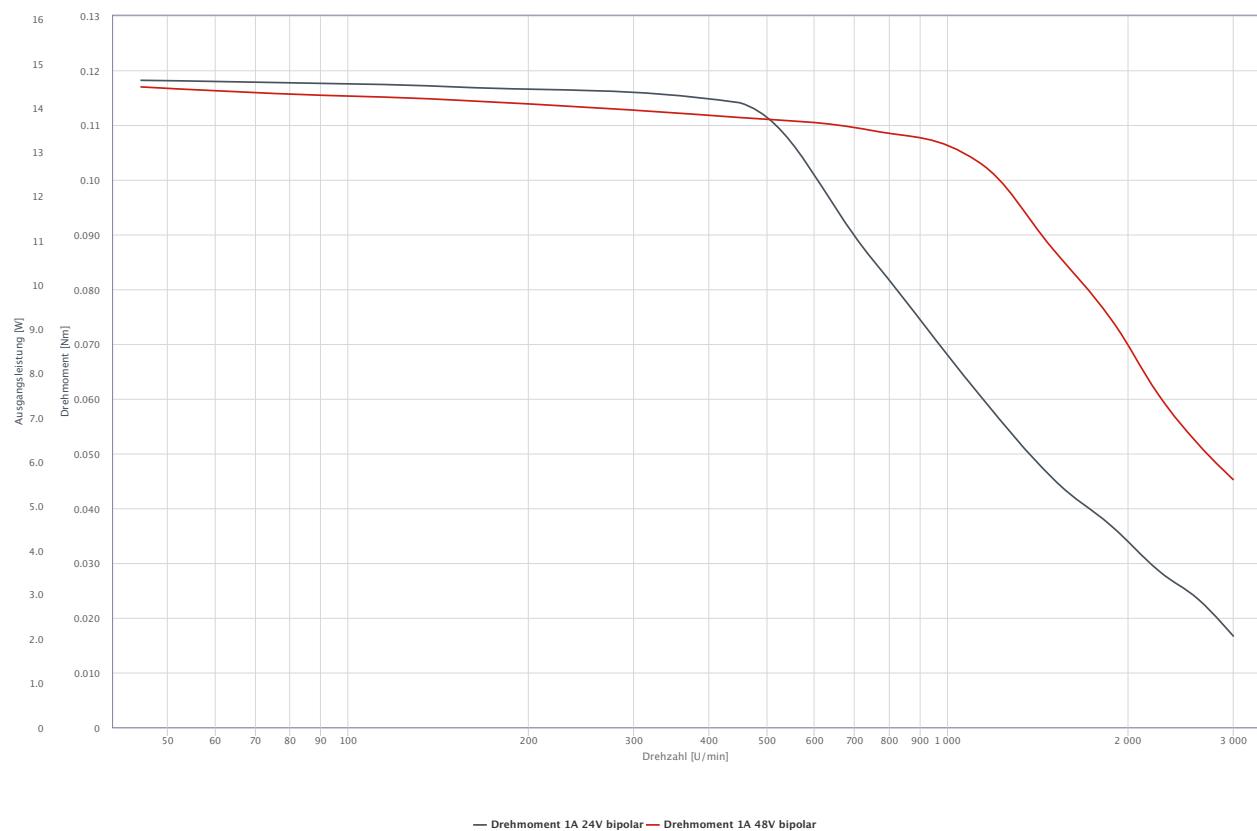
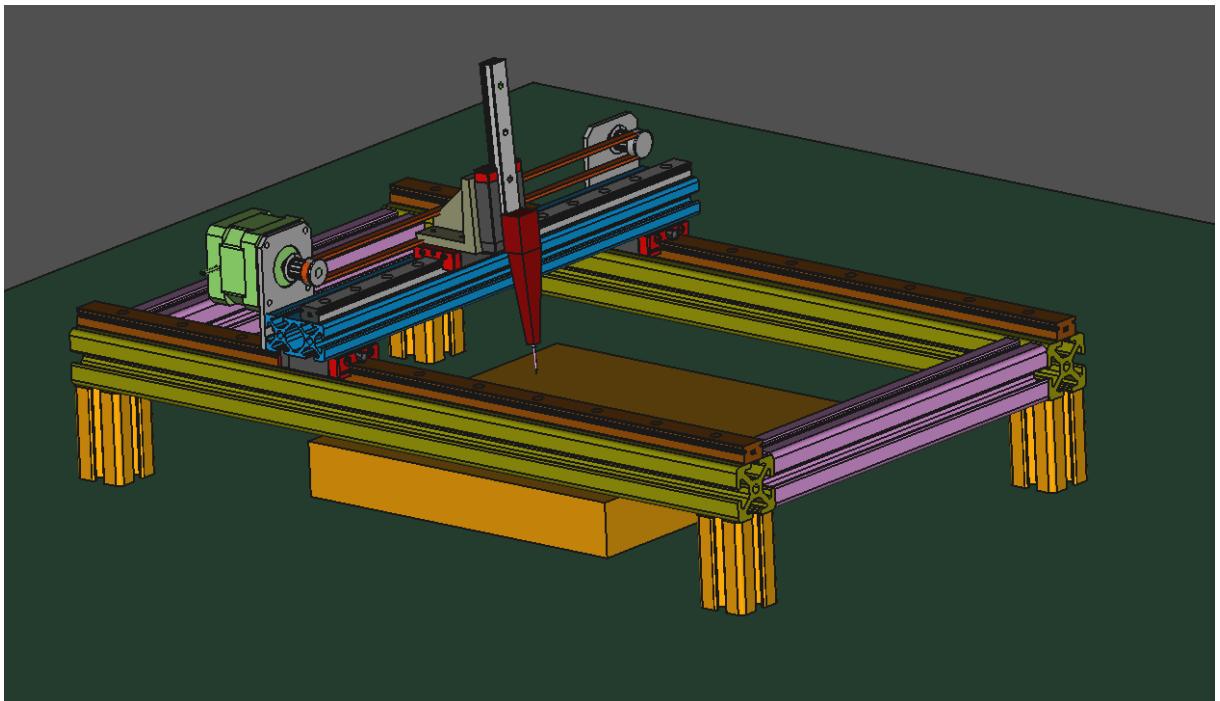
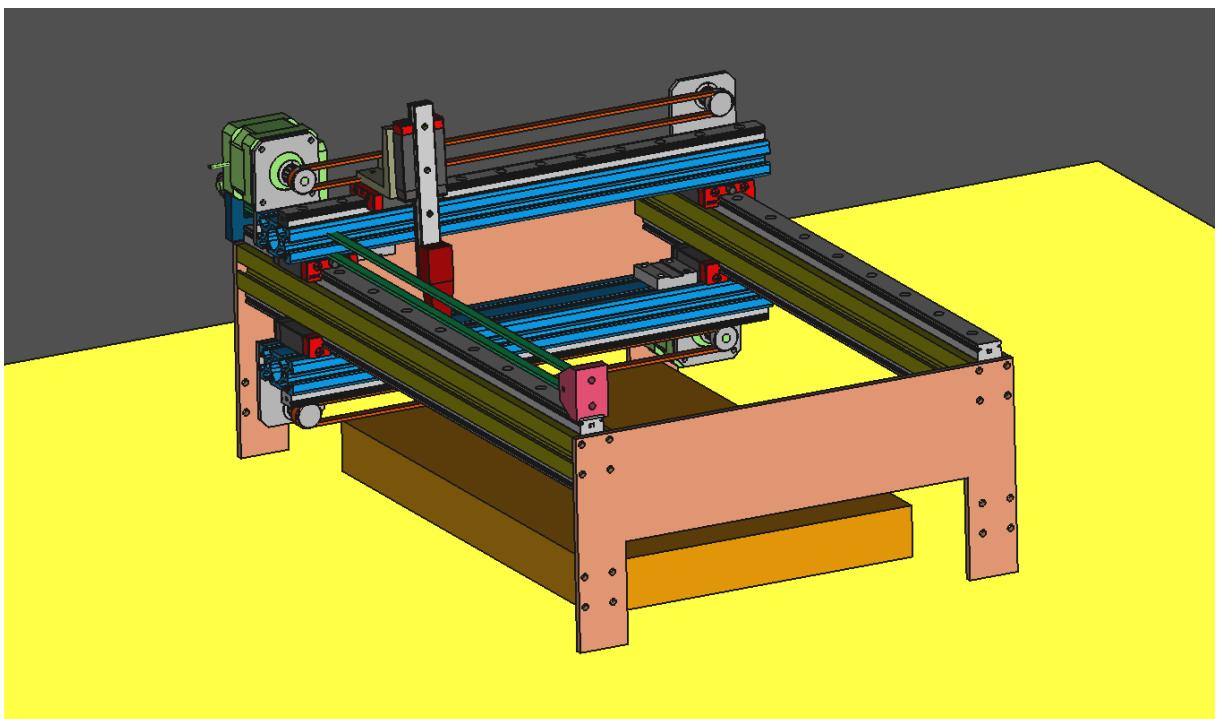


Abbildung C.1.: Quelle: [68]

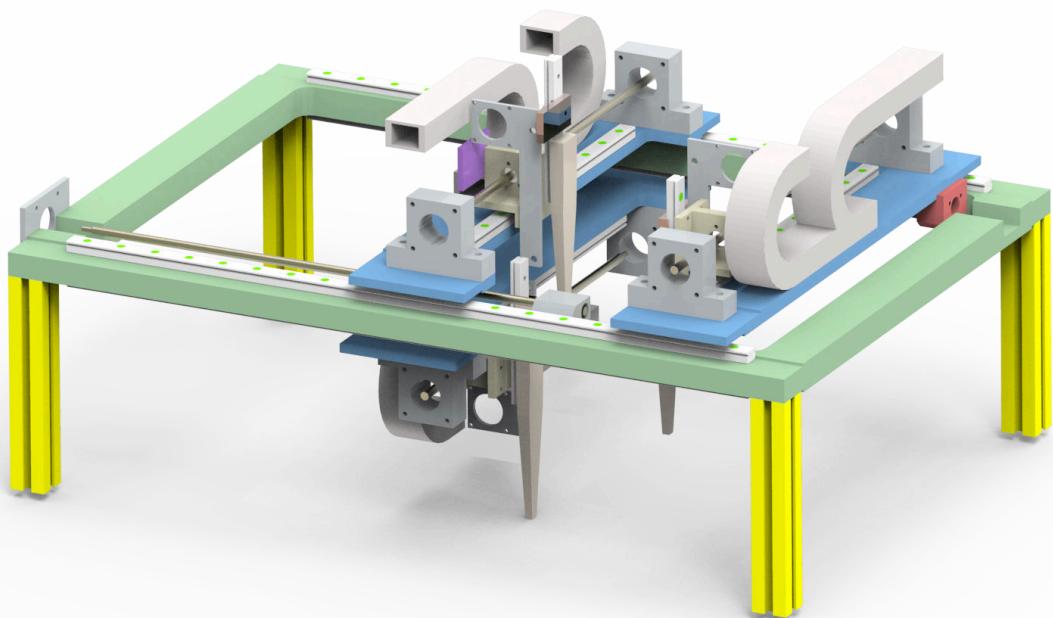
D. Entwürfe der Kontaktierhardware



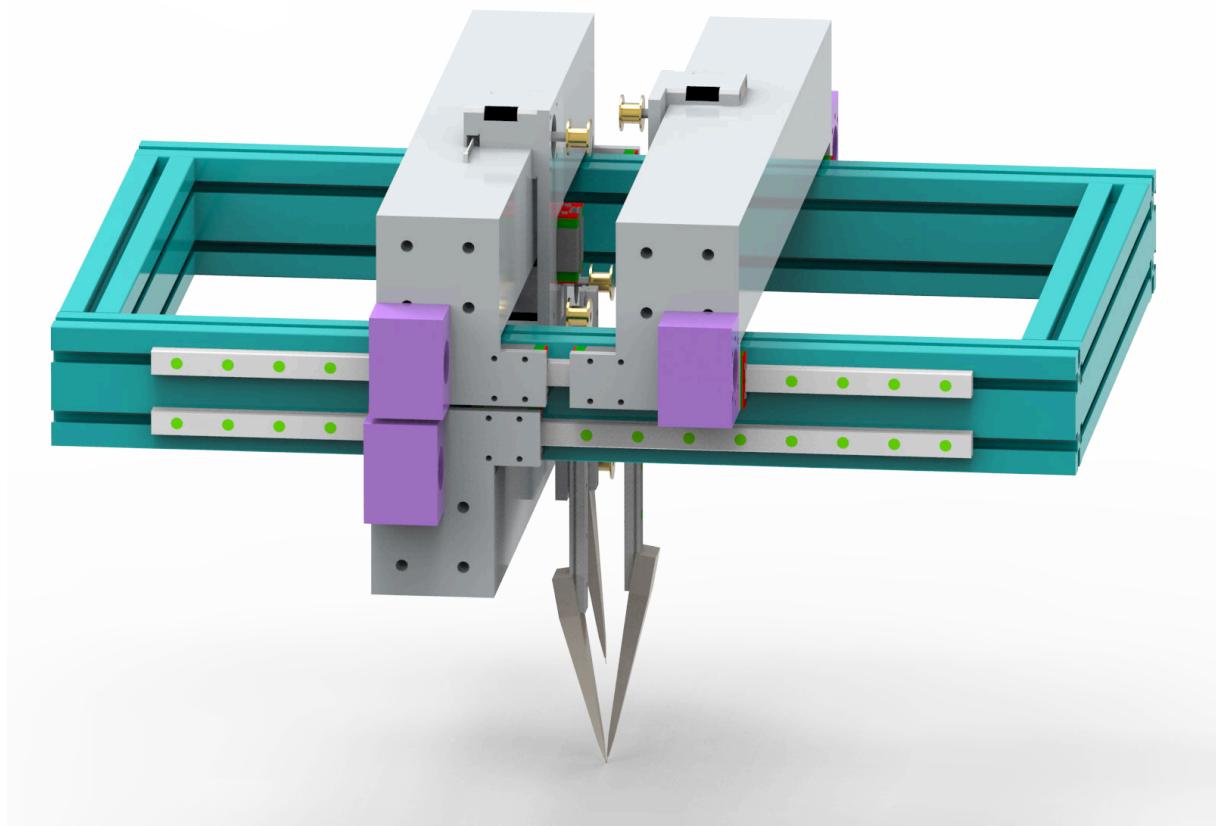
(a) Design 1.1



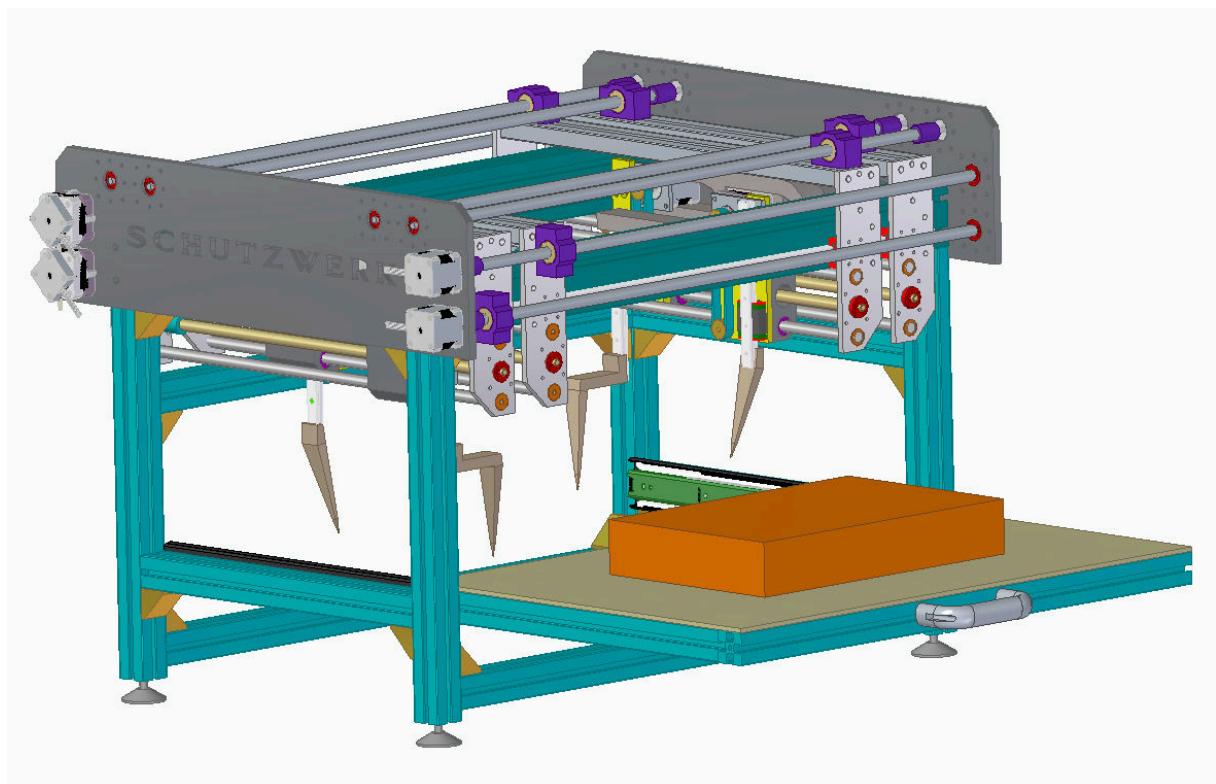
(b) Design 1.2



(c) Design 2.0



(d) Design 3.0



(e) Design 4.0

Literaturverzeichnis

- [1] A. Shamsoshoara, A. Korenda, F. Afghah und S. Zeadally, „A Survey on Hardware-based Security Mechanisms for Internet of Things“, *CoRR*, Jg. abs/1907.12525, 2019.
arXiv: 1907.12525. Adresse: <http://arxiv.org/abs/1907.12525>.
- [2] BMBF. (17. Feb. 2019). Projekt SecForCARs - Sicherheit für vernetzte, autonome Fahrzeuge, Adresse: <https://www.forschung-it-sicherheit-kommunikationssysteme.de/projekte/sicherheit-fuer-vernetzte-autonome-fahrzeuge>.
- [3] K. T. Ng, *PCB-RE - Tools & Techniques*. CreateSpace Independent Publishing Platform, 2017, ISBN: 978-1-979-33138-8.
- [4] X. Wu, D. Sahoo und S. C. H. Hoi, „Recent Advances in Deep Learning for Object Detection“, *arXiv e-prints*, arXiv:1908.03673, arXiv:1908.03673, Aug. 2019.
arXiv: 1908.03673 [cs.CV].
- [5] M. Berger, *Test- und Prüfverfahren in der Elektronikfertigung: Vom Arbeitsprinzip bis zu Design für Test-Regeln*. Vde Verlag GmbH, 2012, ISBN: 978-3-8007-3233-3.
- [6] B. Johnson. (2014). EE368: Reverse Engineering of Printed Circuit Boards, Adresse: https://stacks.stanford.edu/file/druid:np318ty6250/Johnson_Reverse_Engineering_PCBs.pdf.
- [7] S. C. Tan und S. T. W. Kit,
„Fast retrievals of test-pad coordinates from photo images of printed circuit boards“, in *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)*, Nov. 2016, S. 464–467. DOI: 10.1109/ICAMechS.2016.7813492.
- [8] R. C. Mata, S. Azmib, R. Daudc, A. N. Zulkiflid und F. K. Ahmade, „Reverse engineering for obsolete single layer printed circuit board (PCB)“, in *2006 International Conference on Computing Informatics*, Juni 2006, S. 1–7. DOI: 10.1109/ICOICI.2006.5276552.
- [9] S. Kleber, H. F. Nölscher und F. Kargl, „Automated PCB Reverse Engineering“, in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, Vancouver, BC: USENIX Association, 2017.
Adresse: <https://www.usenix.org/conference/woot17/workshop-program/presentation/kleber>.

- [10] R. B. Girshick, J. Donahue, T. Darrell und J. Malik, „Rich feature hierarchies for accurate object detection and semantic segmentation“, *CoRR*, Jg. abs/1311.2524, 2013. arXiv: 1311.2524. Adresse: <http://arxiv.org/abs/1311.2524>.
- [11] R. B. Girshick, „Fast R-CNN“, *CoRR*, Jg. abs/1504.08083, 2015. arXiv: 1504.08083. Adresse: <http://arxiv.org/abs/1504.08083>.
- [12] S. Ren, K. He, R. B. Girshick und J. Sun, „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks“, *CoRR*, Jg. abs/1506.01497, 2015. arXiv: 1506.01497. Adresse: <http://arxiv.org/abs/1506.01497>.
- [13] J. Redmon, S. K. Divvala, R. B. Girshick und A. Farhadi, „You Only Look Once: Unified, Real-Time Object Detection“, *CoRR*, Jg. abs/1506.02640, 2015. arXiv: 1506.02640. Adresse: <http://arxiv.org/abs/1506.02640>.
- [14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu und A. C. Berg, „SSD: Single Shot MultiBox Detector“, *CoRR*, Jg. abs/1512.02325, 2015. arXiv: 1512.02325. Adresse: <http://arxiv.org/abs/1512.02325>.
- [15] C. Pramerdorfer und M. Kampel, „A dataset for computer-vision-based PCB analysis“, in *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, Mai 2015, S. 378–381. DOI: 10.1109/MVA.2015.7153209.
- [16] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn und A. Zisserman, „The Pascal Visual Object Classes Challenge: A Retrospective“, *International Journal of Computer Vision*, Jg. 111, Nr. 1, S. 98–136, Jan. 2015.
- [17] Z. Zhao, P. Zheng, S. Xu und X. Wu, „Object Detection with Deep Learning: A Review“, *CoRR*, Jg. abs/1807.05511, 2018. arXiv: 1807.05511. Adresse: <http://arxiv.org/abs/1807.05511>.
- [19] K. Simonyan und A. Zisserman, „Very Deep Convolutional Networks for Large-Scale Image Recognition“, in *International Conference on Learning Representations*, 2015.
- [20] B. Polyak, „Some methods of speeding up the convergence of iteration methods“, *Ussr Computational Mathematics and Mathematical Physics*, Jg. 4, S. 1–17, Dez. 1964. DOI: 10.1016/0041-5553(64)90137-5.
- [21] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu und Xiaoqiang Zheng,

- TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*,
Software available from tensorflow.org, 2015.
Adresse: <https://www.tensorflow.org/>.
- [22] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama und K. Murphy, „Speed/accuracy trade-offs for modern convolutional object detectors“, *CoRR*, Jg. abs/1611.10012, 2016.
arXiv: 1611.10012. Adresse: <http://arxiv.org/abs/1611.10012>.
- [23] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár und C. L. Zitnick, „Microsoft COCO: Common Objects in Context“, *CoRR*, Jg. abs/1405.0312, 2014.
arXiv: 1405.0312. Adresse: <http://arxiv.org/abs/1405.0312>.
- [24] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang und C. Liu, „A Survey on Deep Transfer Learning“, *CoRR*, Jg. abs/1808.01974, 2018.
arXiv: 1808.01974. Adresse: <http://arxiv.org/abs/1808.01974>.
- [25] G. Ritter und J. Wilson, *Handbook of Computer Vision Algorithms in Image Algebra*. Jan. 2001, ISBN: 978-0-8493-0075-2.
- [26] S. Torbert, *Applied Computer Science*. Springer, 2016, ISBN: 978-3-319-30864-7.
- [27] opencv dev team. (9. Sep. 2019). OpenCV 2.4.13.7 documentation - Miscellaneous Image Transformations. Englisch,
Adresse: https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html%5C#floodfill (besucht am 09.09.2019).
- [28] C. Tomasi und R. Manduchi, „Bilateral Filtering for Gray and Color Images“, in *Proceedings of the Sixth International Conference on Computer Vision*, Ser. ICCV '98, Washington, DC, USA: IEEE Computer Society, 1998, S. 839–, ISBN: 81-7319-221-9.
Adresse: <http://dl.acm.org/citation.cfm?id=938978.939190>.
- [29] G. Bradski, „The OpenCV Library“, *Dr. Dobb's Journal of Software Tools*, 2000.
- [30] S. Suzuki und K. Abe, „Topological structural analysis of digitized binary images by border following“, *Computer Vision, Graphics, and Image Processing*, Jg. 30, Nr. 1, S. 32–46, 1985, ISSN: 0734-189X.
DOI: [https://doi.org/10.1016/0734-189X\(85\)90016-7](https://doi.org/10.1016/0734-189X(85)90016-7). Adresse: <http://www.sciencedirect.com/science/article/pii/0734189X85900167>.
- [31] C. Szegedy, S. Ioffe und V. Vanhoucke, „Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning“, *CoRR*, Jg. abs/1602.07261, 2016.
arXiv: 1602.07261. Adresse: <http://arxiv.org/abs/1602.07261>.

- [32] K. He, X. Zhang, S. Ren und J. Sun, „Deep Residual Learning for Image Recognition“, *CoRR*, Jg. abs/1512.03385, 2015.
arXiv: 1512.03385. Adresse: <http://arxiv.org/abs/1512.03385>.
- [33] C. Pramerdorfer und M. Kampel, „PCB Recognition using Local Features for Recycling Purposes“, *VISAPP 2015 - 10th International Conference on Computer Vision Theory and Applications; VISIGRAPP, Proceedings*, Jg. 3, S. 71–78, Jan. 2015.
DOI: 10.5220/0005289200710078.
- [38] I. Loshchilov und F. Hutter, „SGDR: Stochastic Gradient Descent with Restarts“, *CoRR*, Jg. abs/1608.03983, 2016.
arXiv: 1608.03983. Adresse: <http://arxiv.org/abs/1608.03983>.
- [39] K. Sato, C. Young und D. Patterson. (12. Mai 2017). An in-depth look at Google's first Tensor Processing Unit (TPU),
Adresse: <https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu>.
- [40] R. Padilla, *Object-Detection-Metrics*,
<https://github.com/rafaelpadilla/Object-Detection-Metrics>, 2019.
- [41] Z. Li und F. Zhou, „FSSD: Feature Fusion Single Shot Multibox Detector“, *CoRR*, Jg. abs/1712.00960, 2017.
arXiv: 1712.00960. Adresse: <http://arxiv.org/abs/1712.00960>.
- [42] J. Jeong, H. Park und N. Kwak, „Enhancement of SSD by concatenating feature maps for object detection“, *CoRR*, Jg. abs/1705.09587, 2017.
arXiv: 1705.09587. Adresse: <http://arxiv.org/abs/1705.09587>.
- [43] K. He, G. Gkioxari, P. Dollár und R. B. Girshick, „Mask R-CNN“, *CoRR*, Jg. abs/1703.06870, 2017.
arXiv: 1703.06870. Adresse: <http://arxiv.org/abs/1703.06870>.
- [44] K. Carlson und B. Hauptmann, „Reverse engineering for board test“, in *2009 IEEE AUTOTESTCON*, Sep. 2009, S. 224–228.
DOI: 10.1109/AUTEST.2009.5314092.
- [48] R. Szeliski, *Computer Vision - Algorithms and Applications*, 2011. Aufl.
Berlin Heidelberg: Springer Science & Business Media, 2010, Kap. 2,
ISBN: 978-1-848-82935-0.
- [49] H.-J. Gevatter und U. Grünhaupt,
Handbuch der Mess- und Automatisierungstechnik in der Produktion, 2. Aufl.
Springer, 2006, ISBN: 3-540-21207-8.
- [50] M. Weck, *Werkzeugmaschinen 2: Konstruktion und Berechnung*. Springer Vieweg, 2005,
ISBN: 978-3-540-22502-7.

- [54] RepRap, *RepRap Options*, [Online; accessed 11-September-2019], 14. Aug. 2019.
Adresse: https://reprap.org/mediawiki/index.php?title=RepRap_Options&oldid=185981.
- [55] H. Wittel, D. Muhs, D. Jannasch und J. Voßiek,
Roloff/Matek Maschinenelemente - Normung, Berechnung, Gestaltung, 19. Aufl.
Wiesbaden: Vieweg+Teubner, 2009, ISBN: 978-3-8348-0689-5.
- [56] M. Weck, *Werkzeugmaschinen 4: Automatisierung von Maschinen und Anlagen*.
Springer Vieweg, 2006, ISBN: 978-3-540-22507-2.
- [60] ——, *Werkzeugmaschinen 5 : Messtechnische Untersuchung und Beurteilung, dynamische Stabilität*. Berlin: Springer, 2006, ISBN: 9783642387487.
- [61] G. Facciolo, C. D. Franchis und E. Meinhardt-Llopis,
„Automatic 3D Reconstruction from Multi-date Satellite Images“, in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Juli 2017, S. 1542–1551. DOI: 10.1109/CVPRW.2017.198.
- [62] E. Marchand, H. Uchiyama und F. Spindler, „Pose Estimation for Augmented Reality: A Hands-On Survey“, *IEEE Transactions on Visualization and Computer Graphics*, Jg. 22, Nr. 12, S. 2633–2651, Dez. 2016. DOI: 10.1109/TVCG.2015.2513408.
- [63] Z. Zhang, „A flexible new technique for camera calibration“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 22, Nr. 11, S. 1330–1334, Nov. 2000, ISSN: 0162-8828. DOI: 10.1109/34.888718.
- [65] D. W. Marquardt, „An Algorithm for Least-Squares Estimation of Nonlinear Parameters“, *Journal of the Society for Industrial and Applied Mathematics*, Jg. 11, Nr. 2, S. 431–441, 1963, ISSN: 03684245.
Adresse: <http://www.jstor.org/stable/2098941>.
- [66] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney und D. Sorensen, *LAPACK Users' Guide*, Third. Philadelphia, PA: Society for Industrial und Applied Mathematics, 1999, ISBN: 0-89871-447-8 (paperback).
- [67] W. T. Vetterling, S. A. Teukolsky, W. H. Press und B. P. Flannery, *Numerical Recipes in C - The Art of Scientific Computing*, 2. Aufl.
Cambridge: Cambridge University Press, 1992, ISBN: 978-0-521-43720-2.

Bildquellen

- [18] (21. Aug. 2019). CNN architecture,
Adresse: https://miro.medium.com/proxy/1*N4h1SgwbWNmtrRhszM9EJg.png.
- [34] (13. März 2019). Ergebnis Google Bildersuche, Suchstring "microcontroller",
Adresse: <https://upload.wikimedia.org/wikipedia/commons/thumb/1/18/PIC18F8720.jpg/220px-PIC18F8720.jpg>.
- [35] (13. März 2019). Ergebnis Google Bildersuche, Suchstring "pcb",
Adresse: http://nepsu.com/blog/img/nepsu_pcbserviceguide/cover.jpeg.
- [36] (13. März 2019). Ergebnis Google Bildersuche, Suchstring "pcb red",
Adresse: https://ae01.alicdn.com/kf/HTB1J515KVXXXXbbXFXXq6xFXXXh/New-arrival-CGA-TO-VGA-converter-CGA-EGA-YUV-to-VGA-PCB-Red-2-VGA-output.jpg_640x640.jpg.
- [37] (13. März 2019). Ergebnis Google Bildersuche, Suchstring "platine",
Adresse: <https://www.robotico.de/bilder/produkte/gross/Husqvarna-Platine-fuer-Ladestation-fuer-AM-305-308.jpg>.
- [45] SPEA. (5. März 2019). SPEA 4080, Adresse: https://www.spea-te.de/fileadmin/_processed/_csm_4080_ba48ca2ca4.jpg.
- [46] PRUSA Research. (5. März 2019). Original Prusa i3 MK3S kit,
Adresse: https://shop.prusa3d.com/fotky/COVER_GENERATOR_MK3.jpg.
- [47] UFACTORY. (5. März 2019). uArm Swift Pro,
Adresse: <https://www.ufactory.cc/static/img/banner3.ebe0f49.jpg>.
- [51] DOLD Mechatronik. (29. Aug. 2019). Linearführungsset 94393, Adresse:
<https://www.dold-mechatronik.de/media/image/product/45341/lg/set-4x-vollkunststofflager-rjmp-01-20-mit-alugehaeuse-sce20---2x-praezisionswellen-20mm-h6-geschliffen-und-gehaertet-100mm-mit-gewindebohrungen-m10x25.jpg>.
- [52] ——, (29. Aug. 2019). Supported Rails Set 94317, Adresse:
<https://www.dold-mechatronik.de/media/image/product/45371/lg/set-4x-linearlager-tbr20uu---2x-supported-rail-tbs20-400mm.jpg>.

- [53] ——, (29. Aug. 2019). Profilschienenset 78669, Adresse: <https://www.dold-mechatronik.de/media/image/product/50195/lg/linearwagen-mr-12-mn-zz-blockmodell-inkl--schmiersystem.jpg>.
- [57] Nanotec. (30. Aug. 2019). Produktbild ST3518M1004-A - Schrittmotor - NEMA 14, Adresse: https://de.nanotec.com/fileadmin/_processed_/a/7/csm_Schrittmotor_ST3518M1004_c46a45f36f.jpg.
- [58] TRINAMIC. (30. Aug. 2019). Produktbild TMC2130, Adresse: https://www.trinamic.com/fileadmin/_processed_/0/9/csm_tmc2130-LA_3808873335.jpg.
- [59] BIGTREETECH. (30. Aug. 2019). SKR V1.3 Top, Adresse: <https://github.com/bigtreetech/BIGTREETECH-SKR-V1.3/blob/master/Wiring%20diagram/SKR%20V1.3%20Product%20Top%20Picture.jpg>.
- [64] (18. Aug. 2019). Pinhole camera model, Adresse: https://docs.opencv.org/2.4/_images/pinhole_camera_model.png.
- [68] Nanotec. (19. Juni 2019). Motorkennline ST3518M1004-A - Schrittmotor - NEMA 14, Adresse: <https://de.nanotec.com/produkte/446-st3518m1004-a/>.