

ERWEITERUNG EINES KI-GESTÜTZTEN ASSISTENZSYSTEMS ZUR OPTIMIERUNG VON LASERSCHNEIDPARAMETERN FÜR EDELSTAHLBLECHE

Projektarbeit T2000

des Studienganges Elektrotechnik
Fachrichtung Automation
an der Dualen Hochschule Baden-Württemberg
Standort Stuttgart

Tudor Lupsa

08.09.2025

Bearbeitungszeitraum	02.06.25 - 08.09.25
Matrikelnummer, Kurs	1491114, TEL23GR3
Dualer Partner	TRUMPF SE+Co.KG, Ditzingen
Betreuer des Dualen Partners	Manuel Geiger

Sperrvermerk

Die vorliegende Projektarbeit beinhaltet interne und vertrauliche Informationen der Firma TRUMPF SE + Co. KG. Die Weitergabe des Inhalts, der Arbeit im Gesamten oder in Teilen, sowie Anfertigen von Kopien oder Abschriften, auch in digitaler Form, sind grundsätzlich untersagt. Ausnahmen bedürfen der schriftlichen Genehmigung durch Herrn Manuel Geiger, Betreuer dieser Projektarbeit bei TRUMPF SE + Co. KG Ditzingen.

Dieser Sperrvermerk gilt zum 28. August 2025.

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Projektarbeit selbstständig und nur unter Verwendung der von mir angegebenen Quellen und Hilfsmittel verfasst zu haben. Sowohl inhaltlich als auch wörtlich entnommene Inhalte wurden als solche kenntlich gemacht. Die Arbeit hat in dieser oder vergleichbarer Form noch keinem anderem Prüfungsgremium vorgelegen.

Datum: _____ Unterschrift: _____

Kurzreferat

Abstract

Inhaltsverzeichnis

Sperrvermerk	I
Eidesstattliche Erklärung	II
Kurzreferat	III
Abstract	IV
Abkürzungsverzeichnis	VI
Abbildungsverzeichnis	VII
Tabellenverzeichnis	VIII
Quellcodeverzeichnis	IX
1 Einführung	1
1.1 Zielsetzung	1
1.2 Vorgehensweise	1
2 Grundlagen	3
2.1 Laserlichtschneiden	3
2.2 Grat und Rauheit eines Blechteils	4
2.3 Convolutional Neural Networks (CNNs)	5
3 Stand der Technik	7
3.1 KI-gestütztes Laserschneiden - Cutting Assistant	7
3.2 3D-Messsystem - Keyence	7
4 Identifikation der Schnittabrißgrenze und Datengenerierung	8
5 Messmethodik und Datenerfassung in der Messzelle	10
5.1 Anpassung des Handscanner-Setups	13
5.2 Optimierung der Vektorberechnung aus 3D-Punktwolken	16
5.2.1 Outlier Detection/Correction des Gratverlaufs	17
5.2.2 Interpolation des Gratverlaufs	19
5.3 Validierung der Messoptimierungen	19
5.3.1 Validierung der verbesserten Ausreißererkennung im Burr-Kanal	19
5.3.2 Validierung der bestehenden Rauheitsanalyse an den neuen Edelsstahldaten	20
6 Reflexion und Ausblick	21
6.1 Reflexion	21
6.2 Ausblick	21
Anhang	22
Literaturverzeichnis	24

Abkürzungsverzeichnis

Ein Abkürzungsverzeichnis ist optional. Das Paket `acronym` kann weit mehr, als hier gezeigt.¹ Beachten Sie allerdings, dass Sie die Einträge selbst in sortierter Reihenfolge angeben müssen.

CRM Customer Relationship Management

DHBW Duale Hochschule Baden-Württemberg

IEEE Institute of Electrical and Electronics Engineers

ITIL IT Infrastructure Library

RoI Return-On-Invest

UCS Universal Character Set

UTF-8 8-Bit UCS Transformation Format

Ergänzende Bemerkung: Eine im Text verwendete Abkürzung sollte bei ihrer ersten Verwendung erklärt werden. Falls Sie sich nicht selbst darum kümmern möchten, kann das das Paket `acronym` übernehmen und auch automatisch Links zum Abkürzungsverzeichnis hinzufügen. Dazu ist an allen Stellen, an denen die Abkürzung vorkommt, `\ac{ITIL}` zu schreiben.

Das Ergebnis sieht wie folgt aus:

- erstmalige Verwendung von `\ac{ITIL}` ergibt: IT Infrastructure Library (ITIL),
- weitere Verwendung von `\ac{ITIL}` ergibt: ITIL

Wo benötigt, kann man mit dem Befehl `\acl{ITIL}` wieder die Langfassung ausgeben lassen: IT Infrastructure Library.

Falls man die Abkürzungen durchgängig so handhabt, kann man durch Paket-Optionen (in `_dhw_praeambel.tex`) erreichen, dass im Abkürzungsverzeichnis nur die tatsächlich verwendeten Quellen aufgeführt werden (Option: `printonlyused`) und zu jedem Eintrag die Seite der ersten Verwendung angegeben wird (Option: `withpage`).

Durch die aktivierte Paket-Option `nohyperlinks` wird verhindert, dass die Einträge im Abkürzungsverzeichnis mit Links zu der Stelle hinterlegt werden, wo der Begriff zum ersten Mal verwendet wird.

¹siehe <http://ctan.org/pkg/acronym>

Abbildungsverzeichnis

1	Schema zur Bestimmung der Rauheit	5
2	Schematischer Aufbau eines CNN	6
3	Handscanner (1), ID-Lesegerät (2), Keyence 3D-Messsystem (3) in der Messzelle	11
4	Smartphone (4) und Industriekamera (5) in der Messzelle	12
5	Vergleich zwischen vorhergesagten und tatsächlichen Werten für den Grat (Burr).	12
6	Vergleich zwischen vorhergesagten und tatsächlichen Werten für die Rauheit (Roughness).	12
7	Handcanner Bild für die Gratschätzung (Baustahl)	14
8	Handcanner Bild für die Gratschätzung (Edelstahl)	14
9	Handscnanner Bild für die Rauheitsschätzung (Baustahl)	14
10	Handscanner Bild für die Rauheitsschätzung (Edelstahl)	14
11	Beispiel-Proben-ID eines Blechstücks aus den Experimentalplänen	15
12	Ausreißererkennung im Burr-Kanal	20

Tabellenverzeichnis

1	Formelzeichen für Rauheit und Grat	4
---	--	---

Quellcodeverzeichnis

5.1	Werkstoffabhängiges Routing der Handscanner-Setups	15
5.2	Outlier Detection/Correction in der Profilverarbeitung (Burr-Kanal)	17
5.3	Pipeline-Parameter Outlier Correction (Edelstahl, Burr-Kanal)	19
5.4	Pipeline-Parameter Outlier Correction (Baustahl, Burr-Kanal)	19
6.1	Interpolation begrenzter Lücken in 1D-Profilen	22
6.2	Achsweise Interpolation von NaNs in Matrizen	23
6.3	Interpolation von X-, Y- und Z-Ebenen	23
6.4	Kapselnde 1D-Pipeline: begrenzte Interpolation mit optionaler Glättung	23
6.5	Schließen kurzer Täler (Valleys) via begrenzter Interpolation	24

1 Einführung

1.1 Zielsetzung

Das Ziel dieser Projektarbeit ist es, ein bestehendes KI-gestütztes Assistenzsystem zu erweitern, das aktuell die Qualität beim Laserschneiden von Baustahlblechen vorhersagt und optimiert. Konkret soll die Leistungsfähigkeit dieses Modells auf Edelstahlbleche übertragen werden. Das aktuelle KI-Modell weist in Bezug auf Edelstahl Defizite auf, da es bisher nur mit Datensätzen von Baustahl trainiert wurde und die spezifischen Eigenschaften von Edelstahl unzureichend berücksichtigt. Die Defizite sind insbesondere in der Vorhersagegenauigkeit der Schnittqualität bei Edelstahlblechen erkennbar. Um diese Lücke zu schließen, sollen neue, speziell auf Edelstahl zugeschnittene Trainingsdaten generiert und in das bestehende Modell integriert werden. Diese Daten erfassen insbesondere typische Eigenschaften wie Schneidgratbildung und Oberflächenrauheit. Zusätzlich werden bestehende optische Messmethoden geprüft und gegebenenfalls angepasst, um ihre Eignung für Edelstahl sicherzustellen. Demnach können die neu generierten Daten vermessen und in die Datenbank für das KI-Modell eingepflegt werden. Das Ziel ist ein robustes und zuverlässiges KI-Modell, das die Qualität von Edelstahlschnitten ebenso präzise vorhersagen kann wie bereits für Baustahl.

1.2 Vorgehensweise

Zunächst wird ein systematischer Testplan erstellt, um wichtige Schneidparameter, insbesondere Laserleistung, Schnittgeschwindigkeit, Gasdruck und Fokuslage, für verschiedene Edelstahldicken zu untersuchen. Ziel dieses Testplans ist es, Parameterbereiche zu identifizieren, die zu sogenannten Schnittabrisse führen. Ein Schnittabriß entsteht, wenn durch ungünstige Schneidparameter das Werkstück nicht vollständig getrennt wird, da die Schnittkante verschweißt.

Nachdem kritische Parameterbereiche, welche zu schlechten Schneidergebnissen führen, ermittelt wurden, werden detaillierte Versuchspläne („Experimentalplans“) entwickelt für die Datengenerierung. Diese umfassen systematische Schneidversuche für Edelstahlbleche mit Dicken bis zu 20 mm. Hierbei werden bewusst sowohl qualitativ hochwertige als auch minderwertige Schneidergebnisse erzeugt. Dies dient dazu, eine umfassende Datengrundlage für die Weiterentwicklung des KI-Modells bereitzustellen.

Die generierten Schneidproben werden anschließend in einer Messzelle vermessen, um die resultierenden Schnittkanten in die KI-Datenbank aufzunehmen. Da die Messmethoden ursprünglich für Baustahl entwickelt wurden, müssen sie für Edelstahl angepasst werden. Dies betrifft insbesondere die Kalibrierung und Einrichtung des Handscanner-Setups, welches Bilder der Schnittkanten

für die Qualitätsschätzung aufnimmt. Ebenso muss die Vektorberechnung des eingesetzten 3D-Punktwolkenscanners optimiert werden. Da Edelstahlschnittkanten typischerweise ausgeprägtere Schneidgrade aufweisen als die von Baustahlblechen, muss die Umrechnung der 3D-Punktwolke in einen 2D-Vektor entsprechend angepasst werden, um die tatsächlichen Merkmale der Schnittkanten präzise abzubilden. Neben der quantitativen Messung des Schneidgrats wird auch die Oberflächenrauheit qualitativ bewertet. Die aufbereiteten Messdaten fließen anschließend in die Erweiterung und das Training des bestehenden KI-Modells ein. Ziel ist, dass dieses Modell anschließend die Qualität der Laserschneidkanten bei Edelstahlblechen zuverlässig vorhersagen kann. Nach erfolgreicher Implementierung erfolgen Validierungstests sowie weitere gezielte Optimierungen, um die Vorhersagequalität kontinuierlich zu verbessern und sicherzustellen, dass das gewählte Parameterset bereits vor dem Schneidprozess zuverlässig bewertet werden kann.

2 Grundlagen

Für die vorliegende Arbeit sind Kenntnisse in den Bereichen Laserlichtschneiden, Blechverarbeitung, Bildverarbeitung und Convolutional Neuroal Networks (CNNs) erforderlich. In diesem Kapitel werden die wichtigsten Grundlagen erläutert.

2.1 Laserlichtschneiden

Beim Laserlichtschneiden wird ein fokussierter Laserstrahl auf die Werkstückoberfläche gerichtet. Das Material schmilzt lokal auf und wird mit einem zugeführten Gas aus dem Schnittspalt ausgeblasen. So entsteht eine schmale, definierte Trennfuge. Die grundlegenden Begriffe und Kenngrößen (z. B. Strahlleistung, Strahlqualität, Fokuslage) sind in (*Optics and photonics — Lasers and laser-related equipment — Vocabulary and symbols* 2018) beschrieben.

In der Produktion kommen vor allem Faser- und Scheibenlaser zum Einsatz. CO₂-Laser werden seltener verwendet. Faser- und Scheibenlaser erlauben hohe Energiedichten im Brennfleck und damit hohe Schnittgeschwindigkeiten, besonders bei dünnen bis mittleren Blechdicken. Die Wahl des Prozessgases beeinflusst Schnittqualität und Kantenfarbe. Beim Schneiden mit Stickstoff bildet sich kaum Oxid an der Schnittkante, die Kante bleibt hell und metallisch. Das ist vorteilhaft, weil in der Regel weniger Nacharbeit erforderlich ist, wie z.B. dem Nachbürsten der Kante, so dass wieder eine metallische Oberfläche zu sehen ist. Beim Schneiden mit Sauerstoff reagiert das Material mit dem Gas und somit entsteht eine dunkle Oxidschicht, die die Kante verfärbt. Der Prozess kann zwar schneller sein, führt aber demnach zu zusätzlicher Nacharbeit, wenn eine helle und einwandfreie Kante gefordert ist.

Die wichtigsten Einstellgrößen dieser sind die Laserleistung, Schnittgeschwindigkeit, Fokuslage und Gasdruck. Eine höhere Leistung erlaubt höhere Geschwindigkeit, bis die Qualitätsgrenzen erreicht sind. Eine unpassende Fokuslage führt schnell zu Grat oder unvollständigem Materialaustrag. Der Gasdruck muss so gewählt werden, dass die Schmelze zuverlässig aus dem Schnittspalt entfernt wird, ohne die Kante aufzurauen. Für die Bewertung der Schnittqualität sind u. a. Grat und Rauheit relevant. ISO 9013 fasst hierfür Klassen und Toleranzen zusammen (*Thermal cutting — Classification of thermal cuts — Geometrical product specification and quality tolerances* 2017).

Für Edelstahl gelten gegenüber Baustahl angepasste Einstellungen. Gründe sind unter anderem eine höhere Reflexion und eine geringere Wärmeabfuhr. In der Praxis bedeutet das, dass die Leistung und die Fokuslage sorgfältig abgestimmt werden müssen. Die Geschwindigkeit muss demnach passend gewählt werden. Als Prozessgas wird meistens Stickstoff eingesetzt, um eine helle Kante und geringen Grat zu erreichen.

2.2 Grat und Rauheit eines Blechteils

Für das Verständis der vorliegenden Arbeit sind Grat und Rauheit wichtige Kenngrößen der Schnittqualität. Dies wird in diesem Abschnitt erläutert.

Grat und Rauheit beschreiben die Qualität der Schnittkante. Der Grat ist ein aufgeworfener Materialrand an der Unterkante und entsteht durch unvollständigen Schmelzaustrag oder Anhaftungen von Schmelze im Schnittspalt. Die Rauheit beschreibt die feinen Höhenabweichungen der Schnittfläche entlang eines Profils $Z(x)$.

Die in Tabelle 1 zusammengefassten Formelzeichen werden in diesem Dokument verwendet.

Tab. 1: Formelzeichen für Rauheit und Grat

Zeichen	Bedeutung	Einheit
$Z(x)$	gefiltertes Profil entlang der Auswertelänge L	μm
Z_i	diskreter Profilwert an Position x_i	μm
L	Auswertelänge	mm
n	Anzahl der Stützstellen in L	–
$X_{s,j}$	j -te Teilstrecke innerhalb von L	mm
R_a	arithmetischer Mittenrauwert	μm
R_z	mittlere Rautiefe aus fünf Teilstrecken	μm
P_j^{\max}, P_j^{\min}	höchster bzw. tiefster Punkt in $X_{s,j}$	μm
h_b	Grathöhe an der unteren Schnittkante	μm

Die Berechnung der Kenngrößen folgt den nachfolgenden Rechenregeln. Der arithmetische Mittenrauwert R_a ist das Mittel der Beträge der Profilabweichung über die Auswertelänge L (*Quick Guide to Surface Roughness Measurement* 2016):

$$R_a = \frac{1}{L} \int_0^L |Z(x)| \, dx$$

und in diskreter Form mit n Stützstellen Z_i :

$$R_a = \frac{1}{n} \sum_{i=1}^n |Z_i|.$$

Die mittlere Rautiefe R_z wird über fünf Teilstrecken $X_{s,1}$ bis $X_{s,5}$ bestimmt (*Surface Roughness Parameters — ISO 4287 overview* 2025). In jeder Teilstrecke wird die Differenz zwischen höchstem und tiefstem Profilpunkt gebildet. R_z ist das arithmetische Mittel dieser fünf Differenzen:

$$R_z = \frac{1}{5} \sum_{j=1}^5 (P_j^{\max} - P_j^{\min}).$$

Die Grathöhe h_b wird als maximale positive Auslenkung des Profils im Randbereich der unteren Schnittkante bestimmt. Grundlage ist dasselbe Profil $Z(x)$ oder ein aus einer Punktfolge abgeleitetes Profil senkrecht zur Kante. Neben h_b können Breite und Form des Grates angegeben werden. Die Profilwerte Z_i stammen aus einer 3D-Punktfolge oder aus einer bildbasierten Profilerfassung. Aus diesen Werten werden R_a und R_z berechnet. Die Grathöhe h_b wird im Kantenbereich aus demselben Profil ermittelt. Einen Überblick zum verwendeten 3D-Messsystem gibt Abschnitt 3.2.

Die Abbildung 1 zeigt das gefilterte Rauheitsprofil $Z(x)$ über der Messstrecke X . Die rote Linie kennzeichnet R_a , Z_i sind die diskreten Profilwerte und $X_{s,j}$ die Teilstrecken für R_z . Die Grathöhe h_b wird an der unteren Schnittkante als größte positive Auslenkung im Kantenfenster bestimmt.

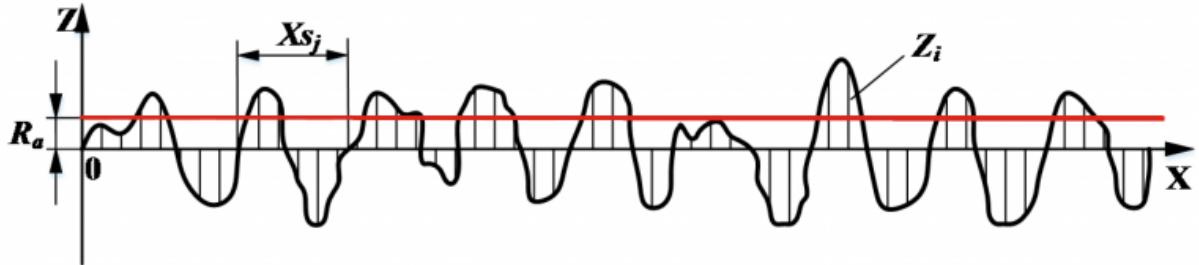


Abb. 1: Schema zur Bestimmung der Rauheit mit Profilwerten Z_i , Teilstrecken $X_{s,j}$ und markiertem R_a .

2.3 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) sind neuronale Netze für Bilddaten. Sie bestehen aus wiederholten Blöcken aus Faltung (Convolution), nichtlinearer Aktivierung (z. B. ReLU) und Subsampling/Pooling sowie einem abschließenden Klassifikations- oder Regressionskopf aus vollverbundenen Schichten. Abbildung 2 zeigt dies am Beispiel eines Roboterbildes. Zu Beginn laufen kleine Filter (typisch 3×3) über das Pixelgitter und reagieren auf lokale Muster. In den ersten Schichten entstehen so Merkmalskarten für einfache Strukturen wie Kanten und Ecken, etwa an der Kontur des Kopfes oder entlang der Armsegmente des Roboters. Eine Aktivierungsfunktion unterdrückt schwache oder negative Antworten und erhöht den Kontrast zwischen relevanten und irrelevanten Bildbereichen. Pooling verdichtet die Merkmalskarten und macht die Darstellung unempfindlicher gegenüber kleinen Verschiebungen. Demnach ist die genaue Position der runden Augen oder Schrauben weniger wichtig als ihr Auftreten.

Mit zunehmender Tiefe kombinieren weitere Faltungen diese einfachen Muster zu komplexeren Teilen wie Gesicht, Gelenken oder ganzen Körpersegmenten. Am Ende wird die verdichtete Repräsentation *geflattet* und von vollverbundenen Schichten zu einer Ausgabe verdichtet, etwa zu Klassenwahrscheinlichkeiten („Roboter ja/nein“) oder zu kontinuierlichen Werten. Das Netzwerk lernt dabei sämtliche Filtergewichte gemeinsam, sodass frühe und späte Schichten aufeinander

abgestimmt sind und eine konsistente Merkmals hierarchie vom Lokalen zum Globalen entsteht (Fischer, Pochwyt 2017).

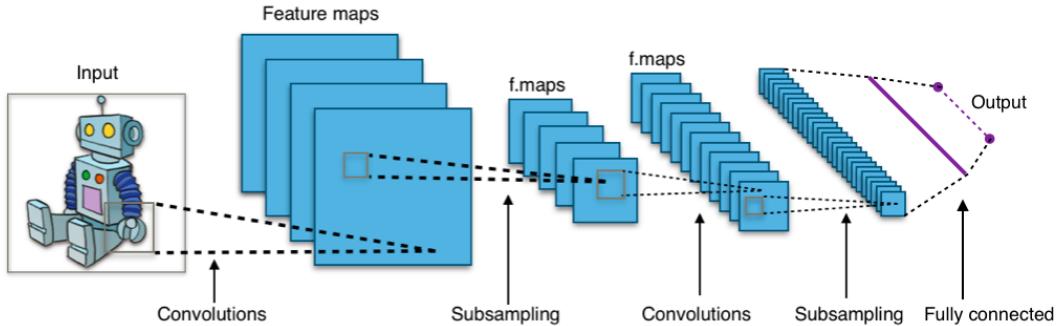


Abb. 2: Schematische Pipeline eines CNN am Beispiel eines Roboterbildes: Faltung extrahiert lokale Muster, Pooling verdichtet die Repräsentation, tiefere Stufen kombinieren Teile zu Objekten, die Entscheidung erfolgt in vollverbundenen Schichten (Fischer, Pochwyt 2017).

Nach demselben Prinzip arbeitet der in dieser Arbeit weiterentwickelte „Cutting Assistent“ (siehe Kapitel 3.1). Frühe Filter reagieren auf Kanten und Texturwechsel in Schnittkantenbildern, Pooling sorgt für Robustheit gegen kleine Lageänderungen, und tiefere Schichten fassen wiederkehrende Muster wie Riefen, Rauheitsstrukturen oder Gratbildung zusammen. Die abschließenden Schichten liefern je nach Aufgabe eine Klassifikation oder Regressionswerte wie Rauheit oder Grathöhe.

3 Stand der Technik

3.1 KI-gestütztes Laserschneiden - Cutting Assistent

3.2 3D-Messsystem - Keyence

4 Identifikation der Schnittabrißsgrenze und Datengenerierung

Ziel dieses Kapitels ist die systematische Abgrenzung des prozesssicheren Arbeitsbereichs beim Laserschneiden von Edelstahlblechen sowie die präzise Identifikation der Parameterbereiche, in denen Schnittabrisse auftreten. Unter einem Schnittabriß wird im Folgenden ein Zustand verstanden, in dem das Werkstück infolge ungeeigneter Parameterkombinationen nicht vollständig getrennt wird, weil der Schnittspalt lokal verschweißt oder die Schmelzaustragung unzureichend ist. Die auf diese Weise gewonnenen Grenzwerte bilden die Grundlage für belastbare Parameterungsempfehlungen und fließen zugleich in die Ausarbeitung strukturierter Versuchspläne zur Datengenerierung ein.

Die Experimente sind als zweidimensionale Rasterstudien ausgelegt, bei denen jeweils zwei der drei wesentlichen Prozessparameter, in dem Fall der Arbeit ist es die Fokuslage, die Schnittgeschwindigkeit und dem Gasdruck, variiert werden, während der dritte Parameter konstant gehalten wird. Die Laserleistung bleibt in diesen Studien konstant. Für jede untersuchte Parameterpaarung wird ein 3×3 -Feld gefertigt, in dem eine Größe entlang der horizontalen und die andere entlang der vertikalen Richtung stufenweise verändert wird. Formal seien die Stufen der beiden variierten Parameter $x \in \{x_1, x_2, x_3\}$ und $y \in \{y_1, y_2, y_3\}$. Der jeweils dritte Parameter z ist auf einem Referenzniveau z_0 fixiert. Die Studie wird sequenziell für alle drei Kombinationen wiederholt, sodass das Prozessfenster in den relevanten Teirläumen konsistent erfasst wird. Die Wahl der Stufen erfolgt material- und dickenspezifisch, aus praxisüblichen Startwerten und internen Erfahrungswerten wird ein plausibler Arbeitsbereich abgeleitet.

Für jede Parameterkombination wird ein Viereck geschnitten. Ein Schnitt gilt als erfolgreich, wenn der Trennschnitt vollständig ist, weder Durchhang noch Wiederaufschmelzen im Schnittspalt beobachtet wird und der Schmelzaustrag kontinuierlich erfolgt. Die Beurteilung erfolgt unmittelbar an der Maschine sowie nachgelagert in der Messzelle durch optische Inspektion und Dokumentation der Schnittkante. Sobald ein erster Grenzbereich identifiziert ist, wird das umliegende Parametergebiet gezielt erkundet, bis ein stabiler Übergang zwischen den Zuständen „Schnitt möglich“ und „Schnittabriß“ reproduzierbar nachgewiesen ist. Die so gewonnenen Grenzpunkte werden im jeweiligen Parameterraum verortet und bilden eine aus Messdaten abgeleitete Näherung des Prozessfensters je Blechdicke.

Auf Basis dieser Grenzanalysen erfolgt die Datengenerierung in Form strukturierter Experimentalpläne. Hierfür wurden insgesamt 17 Edelstahlblechtafeln ($1.5 \text{ m} \times 2 \text{ m}$) in den Dicken 5 mm, 8 mm, 10 mm, 15 mm und 20 mm eingesetzt. Auf jeder Tafel wurden 128 quadratische Proben ($100 \text{ mm} \times 100 \text{ mm}$) geschnitten, sodass ein Gesamtdatenumfang von 2 176 Bauteilen entstand. Die Schneidparameter wurden pro Bauteil innerhalb der vorab definierten, dickenspezifischen Grenzen variiert und nach einem vorgegebenen Schema zufällig ausgewählt. Dieses Design stellt

sicher, dass das Dataset sowohl hochwertige als auch ausdrücklich minderwertige Schneidergebnisse enthält, einschließlich Fehlschnitten und Parameterkombinationen nahe der Schnittabrißgrenze. Solche Negativbeispiele sind für die spätere Modellierung essentiell, um die Trennschärfe zwischen »prozesssicher« und »instabil« zu erhöhen und Fehlklassifikationen zu vermeiden.

Fehlschnitte wurden vollständig protokolliert. Auch wenn betroffene Proben in Einzelfällen nicht aus der Großtafel entnommen werden konnten, gingen diese Versuche mit eindeutiger Kennzeichnung in die Datenbank ein; damit ist bekannt, dass die jeweilige Parameterkombination für die gegebene Blechdicke kein akzeptables Schneidergebnis liefert. Sämtliche entnehmbaren Bauteile werden in der Messzelle vermessen, identifiziert und mit ihren Soll-/Ist-Parametern verknüpft. Die Datenmenge und der Versuchsplan wurden auf Basis der Erfahrungen mit dem bereits für Baustahl trainierten Modell gewählt, um eine ausreichende Abdeckung des Parameterraums und eine robuste Generalisierungsfähigkeit für Edelstahl sicherzustellen.

In Summe ermöglicht die kombinierte Vorgehensweise aus Grenzbestimmung und gezielter Datengenerierung sowohl die belastbare Identifikation der Schnittabrißsgrenzen als auch den Aufbau einer ausgewogenen Datenbasis. Diese bildet die Voraussetzung für die Erweiterung und Validierung des KI-Modells, das künftig die Qualität von Edelstahlschnitten prädiktiv bewerten und prozesssichere Parameterbereiche verlässlich empfehlen soll.

5 Messmethodik und Datenerfassung in der Messzelle

Die Messzelle dient der reproduzierbaren Erfassung aller Messdaten zu den im Rahmen der Experimentalpläne geschnittenen Edelstahlbauteilen. Sie ist als sequenzieller Messprozess ausgelegt, in dem ein mehrachsiger KUKA-Industrieroboter die Proben zwischen den Stationen handhabt. Die Bauteile werden an der Startposition gestapelt bereitgestellt, vom Roboter mittels Vakuumgreifer aufgenommen und der ersten Station zugeführt. Dort erfolgt die automatisierte Probenidentifikation über einen aufgebrachten QR-Code (siehe ID-Lesegerät in Abbildung 3). Die ermittelte Proben-ID wird mit den Metadaten aus den Experimentalplänen (z.B. Blechdicke, Soll-Parameter) verknüpft und dient in der Folge als Schlüssel für die Mess- und Auswertedaten.

Im Anschluss werden an einer Station hochauflöste Aufnahmen der ersten Schnittkante erfasst. Hierzu kommt ein Handscanner (siehe Abbildung 3) zum Einsatz, dessen Aufnahmeparameter, wie z.B. Arbeitsabstand, Belichtung und Auflösung, konstant gehalten werden. Diese Bilddaten bilden die Grundlage für die bildgestützte Qualitätsschätzung des KI-Systems. Ergänzend dazu wird die gleiche Schnittkante mit einem Keyence-3D-Messsystem (siehe Abbildung 3) dreidimensional vermessen, sodass eine 3D-Punktwolke des Kantenverlaufs entsteht. Aus dieser Punktwolke werden definierte Profilverläufe abgeleitet und geometrische Kenngrößen berechnet, die der Erfassung von Gratbildung und der Oberflächenrauheit dienen. Die Berechnung des Grates und der Rauheit ist im obigen Grundlagenkapitel 2.2 näher erläutert. Die genaue funktionsweise des 3D-Messsystems ist im folgenden Kapitel 3.2 ausgiebig erläutert. Die so gewonnenen Ist-Kenngrößen fungieren als Referenz für den späteren Abgleich mit der Bildqualitätsschätzung. In der Abbildung 3 sind die im obigen Abschnitt beschriebenen Komponenten der Messzelle während einer Messung.

In der folgenden Abbildung 3 sind die im obigen Abschnitt beschriebenen Komponenten der Messzelle während einer Messung dargestellt.

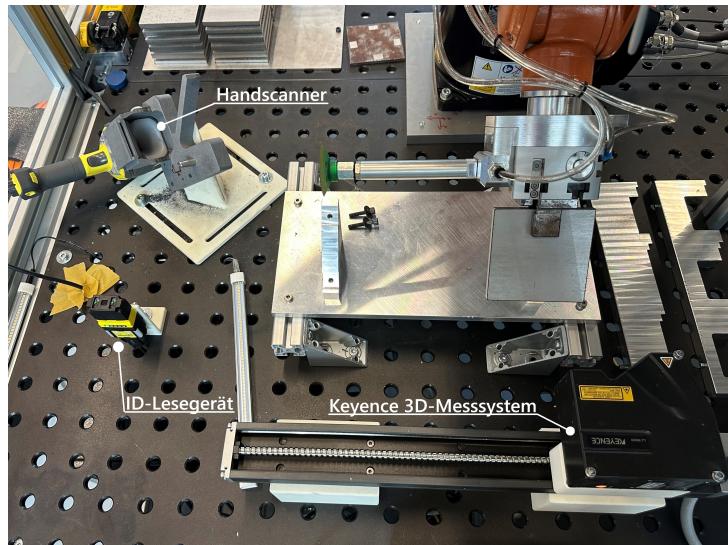


Abb. 3: Handscanner (1), ID-Lesegerät (2), Keyence 3D-Messsystem (3) in der Messzelle

Zur vollständigen Dokumentation werden die Schnittkanten zudem mit einer Industriekamera und einem stationären Smartphone-Setup unter verschiedenen Beleuchtungsbedingungen aufgenommen. Die Kombination aus unterschiedlichen Kameras und Beleuchtungen erhöht die Robustheit der visuellen Beurteilbarkeit und unterstützt die spätere manuelle Nachvollziehbarkeit von Auffälligkeiten. Die erfassten Messdaten des Werkstücks, sowie die daraus abgeleiteten Kenngrößen der Proben-ID zugeordnet und in die zentrale Datenbank überführt.

Die Abbildung 4 zeigt die zuvor beschriebenen Komponenten der Messzelle. Oben im Bild ist das Smartphone (4) mit LED-Ringlicht zu sehen, welches die Schnittkante aus einem flachen Winkel aufnimmt. Unten im Bild ist die Industriekamera (5) mit Auflichtbeleuchtung dargestellt, welche die Schnittkante ebenfalls aus einem flachen Winkel erfasst. Beide Kameras sind fest in der Messzelle montiert und werden automatisch durch den Roboter angesteuert.

Nach der Datenerfassung werden aus der 3D-Messung die tatsächlichen Kenngrößen der Schnittkante berechnet und den Ergebnissen der bildbasierten Qualitätsschätzung gegenübergestellt. Dieser Abgleich ermöglicht die Beurteilung der Übereinstimmung zwischen qualitativer, bildgestützter Bewertung und quantitativer Geometriemessung. Die beschriebenen Messschritte werden für alle vier Schnittkanten jedes Bauteils identisch wiederholt. Abschließend legt der Roboter die vollständig vermessenen Proben an der Endstation geordnet ab.

Anbei ist ein beispielhafter Vergleich der geschätzten Kenngrößen einer Schnittkante im Vergleich zu den gemessenen Kenngrößen in den Abbildungen 5 und 6 dargestellt. Es sind jeweils zwei Diagramme zu sehen, einer für den Gratwert und einer für die Rauheit. Auf den y-Achsen sind die geschätzten Kenngrößen und auf der x-Achse die gemessenen Kenngrößen aufgetragen. Idealerweise liegen alle Punkte auf der Diagonalen, was eine perfekte Übereinstimmung zwischen Schätzung und Messung bedeuten würde. In diesem Beispiel überschätzt das KI-Modell den Rauheitswert leicht, während der Gratwert überwiegend "gut" geschätzt wird. Mit einer leichten

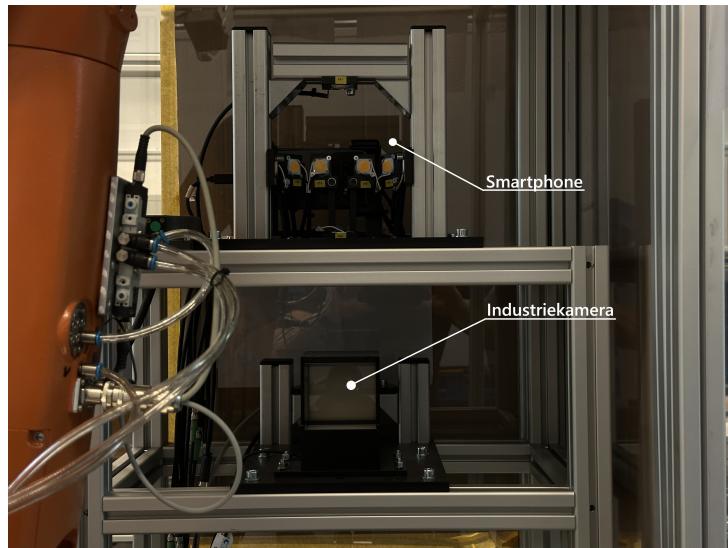


Abb. 4: Smartphone (4) und Industriekamera (5) in der Messzelle

Abweichung ist stets zu rechnen, da die bildbasierte Schätzung eine Näherung darstellt und nicht alle Details der 3D-Messung erfassen kann.

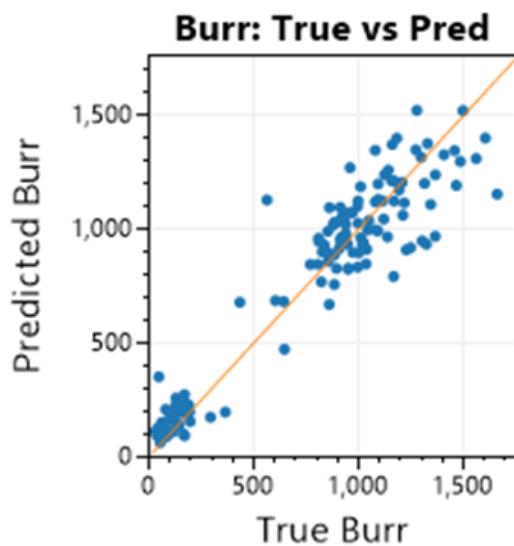


Abb. 5: Vergleich zwischen vorhergesagten und tatsächlichen Werten für den Grat (Burr).

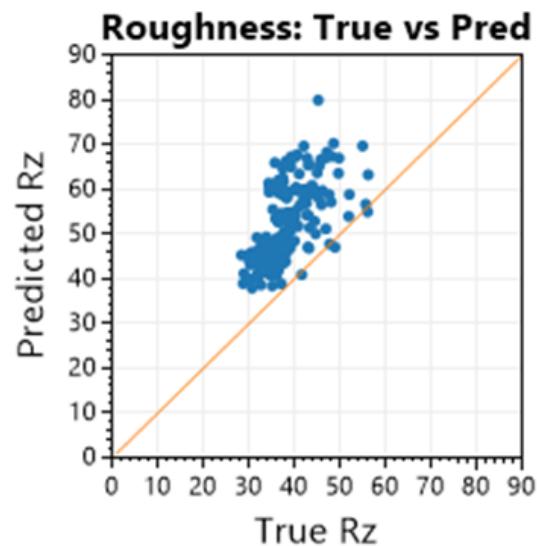


Abb. 6: Vergleich zwischen vorhergesagten und tatsächlichen Werten für die Rauheit (Roughness).

5.1 Anpassung des Handscanner-Setups

Für die bildgestützte Qualitätsschätzung werden die mit dem Handscanner aufgenommenen Schnittkantenbilder als zentrale Eingangsgröße verwendet. Die bisher im Einsatz befindlichen Aufnahmeparameter waren für Baustahl optimiert. Baustahl weist im Vergleich zu Edelstahl eine geringere Oberflächenreflexion und eine tendenziell matte Erscheinung auf. Werden diese Einstellungen unverändert auf Edelstahl angewandt, führt die höhere Reflexion zu Bildartefakten und zu einer unzureichenden Abbildung der relevanten Mikrostruktur (*Stainless Steel 2025*). In der Folge würden Grate (*engl. Burr*) unterrepräsentiert und die Rauheit (*engl. Roughness*) potenziell verfälscht erscheinen. Da die Klassifikation der Bildqualität und die darauf basierende Schätzung von *Burr* und *Roughness* unmittelbar in die Parametrierung des Laserschneidprozesses zurückwirken, ist eine werkstoffabhängige Anpassung des Handscanner-Setups zwingend erforderlich.

Das Aufnahmeprotokoll sieht pro Schnittkante drei Bilder vor: (i) ein bewusst dunkler belichtetes Bild, das primär der Beurteilung der Schnittflächenrauheit dient, sowie (ii) zwei überbelichtete Bilder, die gemeinsam mit dem ersten zu einem HDR-Komposit zusammengeführt werden, um die Kontur und Ausprägung des Grats sicher zu erfassen. Die Kalibrierung der Belichtung erfolgt schrittweise. Zunächst wird die Belichtungszeit für das Rauheitsbild so eingestellt, dass die Textur der Schnittfläche ohne Sättigung und mit klarer Detailzeichnung sichtbar ist. Diese Entscheidung erfolgt in dieser Phase bewusst subjektiv, jedoch anhand vorab definierter visueller Kriterien, wie z.B. ausreichender Tonwertumfang und erkennbarer Strukturkontrast. Im Anschluss werden die Belichtungsparameter der beiden HDR-Bilder iterativ variiert, bis der Grat entlang der Schnittkante über den gesamten Bildbereich eindeutig detektierbar ist, ohne dass umliegende Bereiche vollständig verloren gehen. Da die HDR-Komposition durch die Eingangsbilder beeinflusst wird, erfolgt die Abstimmung der HDR-Belichtungen stets nach der Festlegung des Rauheitsbildes. Die Abbildungen 7, 8, 9 und 10 zeigen exemplarisch die finalen Handscanner Bilder, welche für die Qualitätsschätzung genutzt werden, für Edelstahl im Vergleich zu den bisherigen Bildern für die Qualitätsschätzung von Baustahl. In den Abbildungen ist jeweils die gleiche Schnittkante eines Blechteils aus den geschnittenen Experimentalplänen dargestellt mit einer Dicke von 15 mm.

Zur Sicherstellung der Kompatibilität mit dem bestehenden KI-Modell wird die Anpassung an Referenzaufnahmen aus der bereits validierten Baustahlkonfiguration ausgerichtet. Praktisch bedeutet dies, dass eine Baustahlschnittkante mit den etablierten Baustahleinstellungen aufgenommen wird und die Edelstahlaufnahmen so justiert werden, dass die resultierenden Bildcharakteristika in qualitativer Hinsicht vergleichbar sind. Auf diese Weise wird gewährleistet, dass die Edelstahlbilder in das bestehende Modell eingebunden und mit den vorhandenen Trainings- und Bewertungsroutinen verarbeitet werden können.

Zur konsistenten Anwendung der angepassten Handscanner-Parameter wird das Messzellen-Skript so erweitert, dass das passende Setup automatisiert auf Basis der Bauteilbezeichnung gewählt wird. Die Benennung folgt dem Schema **Maschinename-Experimentalplanname-Material**.

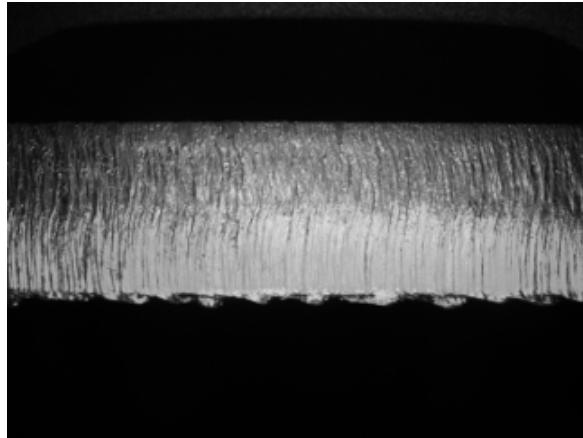


Abb. 7: Handcanner Bild für die Gratschätzung (Baustahl)

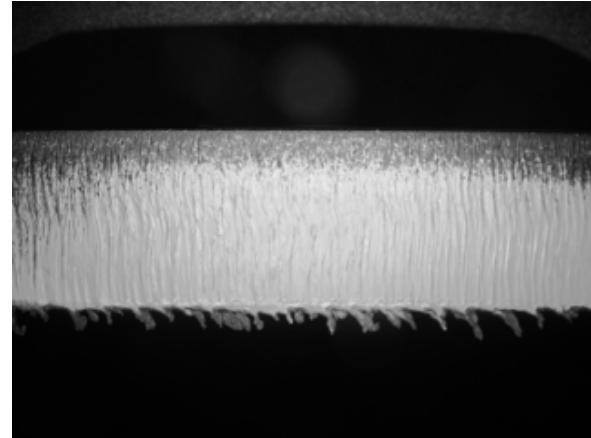


Abb. 8: Handcanner Bild für die Gratschätzung (Edelstahl)

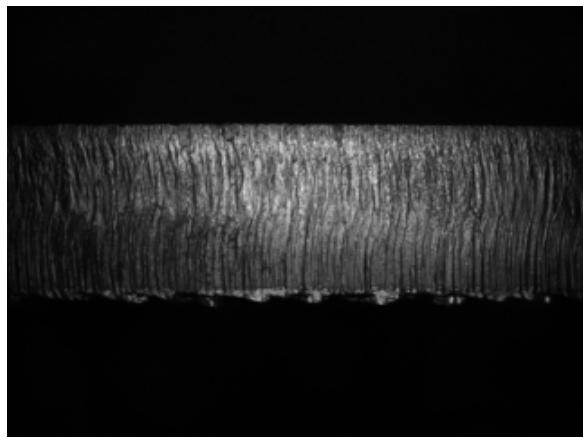


Abb. 9: Handscanner Bild für die Rauheits- schätzung (Baustahl)

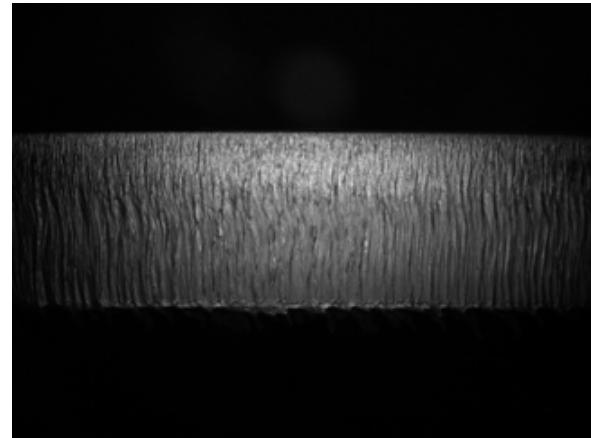


Abb. 10: Handscanner Bild für die Rauheits- schätzung (Edelstahl)

lnummer-Bauteildicke-Bauteilnummer, z.B. A02280E0005-AiMuWrCjd0-3-050-0176. Die folgende Abbildung 11 zeigt eine Beispiel-Proben-ID eines Blechstücks aus den Experimentalplänen mit den einzelnen Segmenten.

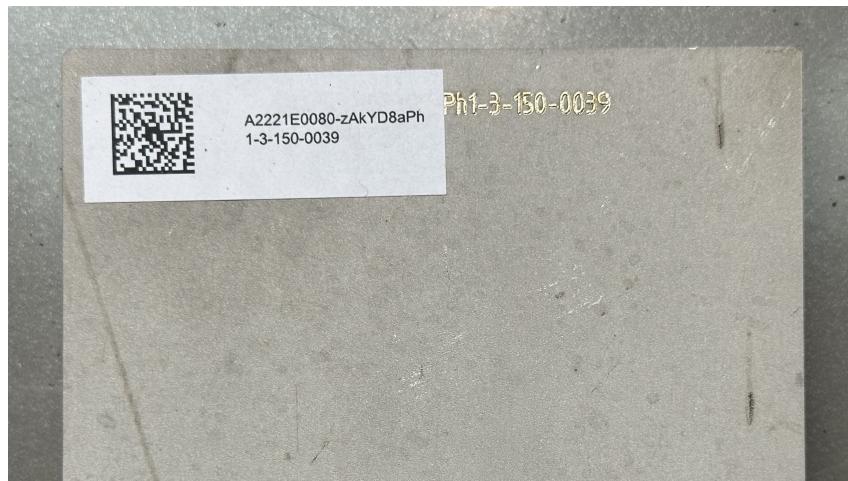


Abb. 11: Beispiel-Proben-ID eines Blechstücks aus den Experimentalplänen

Das Skript parst die Zeichenkette, prüft die Zahl nach dem zweiten Bindestrich und lädt abhängig davon die vordefinierten Handscanner-Einstellungen für den jeweiligen Werkstoff. Auf diese Weise wird sichergestellt, dass die für Edelstahl kalibrierten Belichtungen und Aufnahmeparameter reproduzierbar zur Anwendung kommen und die so erzeugten Bilder ohne systematische Verzerrungen in die Qualitätsmodellierung eingehen. Dies ist im folgendem C-Sharp Quellcode 5.1 dargestellt und im Messzellenskript implementiert.

Quellcode 5.1: Werkstoffabhängiges Routing der Handscanner-Setups

```

1 public static string NameParserMaterial(string input)
2 {
3     if (string.IsNullOrWhiteSpace(input))
4         return "ST";                                // Fallback-Wert
5
6     // Position des ersten und zweiten Bindestrichs ermitteln
7     int firstDash = input.IndexOf('-');
8     int secondDash = firstDash >= 0
9         ? input.IndexOf('-', firstDash + 1)
10        : -1;
11
12    // Prüfen, ob ein zweiter Bindestrich und ein Zeichen dahinter existieren
13    if (secondDash < 0 || secondDash + 1 >= input.Length)
14        return "ST";                                // Fallback-Wert
15
16    char digit = input[secondDash + 1];            // Ziffer einlesen
17
18    // Zuordnung: 1 → SST", 2 → SST", 3 → SSS"(bei Bedarf anpassen)
```

```

19     return digit switch
20     {
21         '1' => "ST",
22         '2' => "ST",
23         '3' => "SS",
24         _ => "ST"           // Default
25     };
26 }
```

5.2 Optimierung der Vektorberechnung aus 3D-Punktwolken

Die mit dem Keyence-System aufgenommene 3D-Punktwolke der Schnittkante bildet die Grundlage für die geometrische Qualitätsauswertung. In der bestehenden Auswertepipeline wird die Punktwolke zunächst segmentiert und in ein lokales Kantenkoordinatensystem überführt. Anschließend erfolgt eine Projektion aus der dreidimensionalen Repräsentation in einen zweidimensionalen Profilverlauf, sodass ein 2D-Vektor entsteht, der den Verlauf des Schneidgrats entlang der Schnittkante beschreibt. Diese Vorgehensweise wurde ursprünglich für Baustahl entwickelt und auf dessen charakteristisch eher wellige, kontinuierliche Gratmorphologie abgestimmt.

Die nachfolgend beschriebenen Anpassungen betreffen ausschließlich den **Grat-/Burr-Kanal** unterteilt in der Ausreißererkennung von Messdaten und der darauffolgenden Interpolation. Für die Rauheitsanalyse werden die in der Baustahl-Pipeline etablierten Einstellungen übernommen. Aus interner Erfahrung zeigt die Rauheitsprüfung eine stabile und konsistente Bewertung.

Bei Edelstahl zeigt sich eine abweichende, ausgeprägt zackige Gratstruktur mit höheren lokalen Gradienten und diskontinuierlichen Profilabschnitten. Die bislang implementierte Outlier-Korrektur ist konzipiert zur Eliminierung sporadischer Messfehler bei Baustahl. Demnach stuft diese den Gratverlauf von Edelstahl fälschlich als Ausreißer ein und glättet sie übermäßig. Dadurch werden relevante Merkmale des Edelstahlgrats unterdrückt und der resultierende Vektorverlauf in Richtung eines künstlich „glatten“ Profils verzerrt.

Erschwerend kommt hinzu, dass im Messprozess partiell überbeschattete Bereiche auftreten können, die vom Sensor nicht erfasst werden. In der bisherigen Pipeline werden solche Lücken durch Interpolation geschlossen, deren Parameter auf die kontinuierlichen Profile von Baustahl zugeschnitten sind. Für den zackigen Edelstahlgrat führt dies zu einer zu starken Annäherung an glatte Zwischenverläufe und damit zu einem Verlust an formcharakteristischer Information.

Zur materialspezifischen Anpassung werden daher zwei Kernmodule überarbeitet. Die Ausreißererkennung Section 5.2.1 mit nachgelagerter Korrektur und die Interpolation Section 5.2.2 fehlender Stützstellen. In der Ausreißererkennung werden die Schwellwerte und die zugrunde liegenden Sensitivitätsmaße an die höhere lokale Krümmung und den gesteigerten Kantenkontrast

des Edelstahlgrats angepasst. Ziel ist eine Differenzierung zwischen echten Messfehlern und materialtypischen Hochfrequenzanteilen. Entsprechend werden Glättungsschritte zurückgenommen, sodass signifikante Gratflanken erhalten bleiben.

Für die Interpolation wird ein konservativer Ansatz gewählt, der Lückenschlüsse bevorzugt entlang lokal konsistenter Nachbarschaften vornimmt und globale, stark glättende Approximationen vermeidet. Damit wird erreicht, dass der rekonstruierte 2D-Vektor fehlende Messpunkte plausibel ergänzt, ohne die charakteristische Zackigkeit des Edelstahlgrats zu nivellieren.

5.2.1 Outlier Detection/Correction des Gratverlaufs

Die Ausreißerbehandlung im **Burr**-Kanal erfolgt zweistufig. Zunächst werden potenzielle Ausreißerpunkte durch den Vergleich des gemessenen Profils mit einem lokal geglätteten Referenzsignal identifiziert. Anschließend werden die dadurch entstehenden Lücken kontrolliert und rekonstruiert. In `outlier_correction_burr` wird das Höhenprofil `z_vec` mittels gleitendem Mittelwert (`np.convolve` mit der Fensterlänge `window`) geglättet. Die absolute Abweichung $\Delta = |z - \bar{z}_{MA}|$ wird punktweise gegen den Schwellwert `threshold` geprüft. Punkte oberhalb des Schwellwertes bilden die Ausreißermaske. Ist der Anteil markierter Punkte größer als `max_nan_values_perc`, wird der Vorgang verworfen (`None`). Andernfalls werden die Ausreißer zu `NaN` gesetzt und mit `smooth_nan_values` rekonstruiert, um numerisches Rauschen zu reduzieren, ohne relevante Strukturen zu nivellieren.

Die Funktion `outlier_correction_profile_lines` setzt denselben Ansatz für einzelne Profilzeilen um, verwendet jedoch standardmäßig einen robusten gleitenden Median (`median=True`) als Referenzsignal. Aus der Abweichung zum Referenzsignal wird eine Ausreißermaske gebildet und über `max_outlier_values_perc` validiert. Markierte Punkte werden zu `NaN` gesetzt und anschließend nur dann interpoliert, wenn die Lückenlänge die Vorgabe `max_gap` nicht überschreitet (`interpolate_limited_nans`). Die Rauheitsanalyse nutzt unverändert die etablierten Baustahl-Einstellungen, hierfür erfolgt keine Umparametrisierung.

Quellcode 5.2: Outlier Detection/Correction in der Profilverarbeitung (Burr-Kanal)

```

1 def outlier_correction_burr(x_vec, z_vec, threshold=0.04, window=20,
2                             ↪ max_nan_values_perc=0.4):
3     """
4     Entfernt Ausreißer anhand eines Moving-Average-Vergleichs.
5     """
6     z_vec = np.copy(z_vec)
7     z_smoothed = np.convolve(z_vec, np.ones(window) / window, mode='same')
8     difference = np.abs(z_vec - z_smoothed)
9
10    outlier_mask = difference > threshold
11    num_outliers = np.sum(outlier_mask)

```

```

11
12     if num_outliers / len(z_vec) > max_nan_values_perc:
13         return None, None
14
15     z_vec[outlier_mask] = np.nan
16     z_vec_clean = smooth_nan_values(x_vec, z_vec)
17
18     return x_vec, z_vec_clean
19
20
21 def outlier_correction_profile_lines(line, outlier_threshold=0.04, window_size=30,
22                                     median=True, max_outlier_values_perc=0.35,
23                                     ↪ max_gap=5):
24     """
25     Entfernt Ausreißer in Höhenprofilen basierend auf Median- oder
26     ↪ Mittelwert-Vergleich.
27     """
28
29     Z_series = pd.Series(line)
30     tmp_line = line.copy()
31
32     if median:
33         moving_avg = Z_series.rolling(window_size, min_periods=5,
34                                     ↪ center=True).median()
35     else:
36         moving_avg = Z_series.rolling(window_size, min_periods=5, center=True).mean()
37
38     difference = np.abs(line - moving_avg)
39     id_outlier = difference > outlier_threshold
40     count_outlier = np.sum(id_outlier)
41
42     if count_outlier / len(line) > max_outlier_values_perc:
43         return None, None
44
45     tmp_line[id_outlier] = np.nan
46     tmp_line = interpolate_limited_nans(tmp_line, max_gap=max_gap)
47
48     return tmp_line, moving_avg

```

In der Edelstahl-Pipeline wurden die Schwellwerte der Burr-Outlier-Korrektur angehoben und das Fenster leicht verkürzt, um zackige, materialspezifische Hochfrequenzanteile nicht fälschlich als Ausreißer zu markieren. Zum Vergleich sind nachfolgend die verwendeten Parameter für Edelstahl sowie die bisherige Baustahl-Konfiguration aufgeführt (siehe auch Quellcode 5.3 und Quellcode 5.4). Für die Rauheitsanalyse wurden keine Parameteränderungen vorgenommen.

Quellcode 5.3: Pipeline-Parameter Outlier Correction (Edelstahl, Burr-Kanal)

```
1 # Parameter burr outlier correction (Edelstahl)
2 burr_outlier_threshold : 0.06 # Threshold for Moving Average Difference Filter [mm]
3 burr_outlier_window     : 9    # Window for Moving Average Difference Filter [samples]
```

Quellcode 5.4: Pipeline-Parameter Outlier Correction (Baustahl, Burr-Kanal)

```
1 # Parameter burr outlier correction (Baustahl)
2 burr_outlier_threshold : 0.03 # Threshold for Moving Average Difference Filter [mm]
3 burr_outlier_window     : 10   # Window for Moving Average Difference Filter [samples]
```

5.2.2 Interpolation des Gratverlaufs

Die Interpolation rekonstruiert fehlende Messwerte (`NaN`), die durch Ausreißerkennzeichnung oder unvollständige Erfassung entstehen. Ziel ist die Wiederherstellung eines plausiblen Gratverlaufs, daher werden nur kurze, lokal begrenzte Lücken gefüllt und größere Ausfälle bleiben markiert. Durch die angepasste Ausreißerkennung werden weniger echte Messpunkte fälschlich als Ausreißer markiert, sodass deutlich weniger Lücken entstehen und nicht mehr unnötig interpoliert wird. Am Interpolationsskript wurden keine Änderungen vorgenommen. Alle Konstanten bleiben unverändert, insbesondere die maximale Lücke sowie Glättungs- und Fensterparameter. Der zugehörige Quellcode ist im Anhang dokumentiert. Die Validierung dessen erfolgt im folgendem Kapitel 5.3.

5.3 Validierung der Messoptimierungen

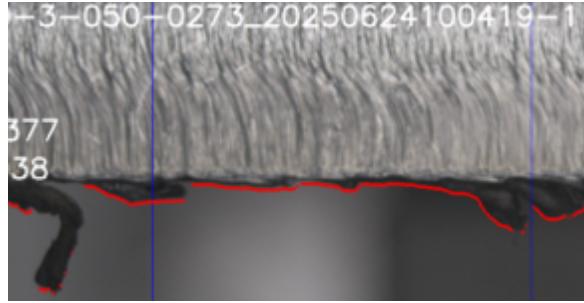
Im folgendem Kapitel werden die verbesserten Messmethoden für die Grat- und Rauheitsanalyse validiert, um bestenfalls die Optimierung für Datenerfassung zu nutzen, so dass das KI-Modell neu trainiert werden kann.

5.3.1 Validierung der verbesserten Ausreißerkennung im Burr-Kanal

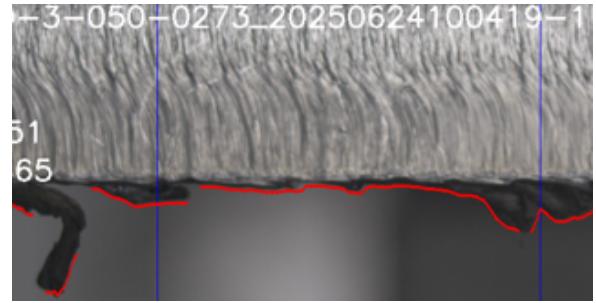
In diesem Abschnitt werden die edelstahl-optimierten Einstellungen der Ausreißerkennung validiert. Verglichen wird die frühere Baustahl-Parametrierung mit der angepassten Erkennung für Edelstahl.

Die Abbildung 12 den Vergleich der verbesserten und der unverbeserten Ausreißerkennung. Hierfür wurden jeweils das gleiche Sektionsbild des gleichen Edelstahlblechs. Die rote Linie ist der abgeleitete Burr-Vektor. Auf dem linken Schaubild ist die unverbeserte Ausreißerkennung erkennbar, welche für Baustahl werdet wird und rechts die verbesserte Ausreißerkennung. Demnach ist der verbesserte Verlauf durchgängig, es entstehen weniger fehlerhafte Lücken und die

interpolierten Abschnitte sind plausibler. Quantitativ sinkt der Burr-Wert der gezeigten Sektion von $681,99 \mu\text{m}$ (Baustahl-Parameter) auf $650,88 \mu\text{m}$ (Edelstahl-Parameter), jedoch erfolgt die einschätzende Bewertung visuell. Es wird geprüft, wie gut die rote Linie dem sichtbaren Gratverlauf folgt.



Unverbesserte Ausreißererkennung: sichtbare Lücken und Fehlverfolgungen.



Verbesserte Ausreißererkennung: durchgängiger Verlauf, plausible Interpolation.

Abb. 12: Links ist die unverbesserte Ausreißererkennung erkennbar, welche für Baustahl werdet wird und rechts die verbesserte Ausreißererkennung. Die rote Kurve ist der abgeleitete Burr-Vektor.

Demnach ist die verbesserte Ausreißererkennung für Edelstahl validiert. Die Rauheitsanalyse wurde nicht verändert, da diese in internen Erfahrungen eine stabile und konsistente Bewertung zeigt.

5.3.2 Validierung der bestehenden Rauheitsanalyse an den neuen Edelsdtahldaten

In diesem Abschnitt wird die Rauheitsanalyse validiert. Die Berechnungsmethode bleibt unverändert. Die Belichtung des Handscanners wurde so angepasst, dass die Struktur der Schnittfläche zuverlässig erfasst wird (siehe Abschnitt 5.1). Zur Prüfung der Eignung wurde die bestehende Rauheitsberechnung auf die neuen Bilder von Edelstahlblechen angewendet und auf mögliche Ausreißer in den resultierenden Rauheitswerten untersucht. Es traten keine systematischen Ausreißer auf, daher wird die Berechnung vorerst unverändert weitergeführt. Anpassungen werden erst nach weiterführenden Modelltrainings geprüft.

6 Reflexion und Ausblick

6.1 Reflexion

6.2 Ausblick

Anhang

Quellcode 6.1: Interpolation begrenzter Lücken in 1D-Profilen

```
1 def interpolate_limited_nans(vector, max_gap=5):
2     """
3         Interpoliert nur kleine NaN-Lücken (<= max_gap) mit weicher Spline-Interpolation.
4     """
5     vector = vector.copy()
6     isnan = np.isnan(vector)
7     indices = np.arange(len(vector))
8
9     if not np.any(~isnan):
10         return None
11
12     nan_groups = []
13     in_nan = False
14     start = 0
15     for i, val in enumerate(isnan):
16         if val and not in_nan:
17             in_nan = True
18             start = i
19         elif not val and in_nan:
20             in_nan = False
21             nan_groups.append((start, i - 1))
22     if in_nan:
23         nan_groups.append((start, len(vector) - 1))
24
25     for start, end in nan_groups:
26         gap_size = end - start + 1
27         if gap_size <= max_gap:
28             left = start - 1
29             right = end + 1
30             if left < 0 or right >= len(vector):
31                 continue
32             if np.isnan(vector[left]) or np.isnan(vector[right]):
33                 continue
34
35             # Verwende CubicSpline statt np.interp
36             x_known = [left, right]
37             y_known = [vector[left], vector[right]]
38             cs = CubicSpline(x_known, y_known, bc_type='natural')
39             interp_indices = indices[start:end+1]
40             vector[start:end+1] = cs(interp_indices)
```

```
41
42     return vector
```

Quellcode 6.2: Achsweise Interpolation von NaNs in Matrizen

```
1 def interpolate_nan(matrix_in, axis=0, max_gap=5):
2     """
3     Interpoliert NaNs in einer Matrix entlang der gegebenen Achse mit Begrenzung.
4     """
5     matrix = matrix_in.copy()
6     x, y = matrix.shape
7
8     if axis == 0:
9         for col_index in range(y):
10            if np.isnan(matrix[:, col_index]).any():
11                vec = interpolate_limited_nans(matrix[:, col_index], max_gap=max_gap)
12                if vec is not None:
13                    matrix[:, col_index] = vec
14    else:
15        for row_index in range(x):
16            if np.isnan(matrix[row_index, :]).any():
17                vec = interpolate_limited_nans(matrix[row_index, :], max_gap=max_gap)
18                if vec is not None:
19                    matrix[row_index, :] = vec
20
21    return matrix
```

Quellcode 6.3: Interpolation von X-, Y- und Z-Ebenen

```
1 def interpolate_planes(X, Y, Z, results_directory=None, max_gap=0.5):
2     """
3     Interpoliert X, Y, Z Ebenen entlang sinnvoller Achsen mit Lückenbegrenzung.
4     """
5     X_interpolated = interpolate_nan(X, axis=1, max_gap=max_gap)
6     Y_interpolated = interpolate_nan(Y, axis=1, max_gap=max_gap)
7     Z_interpolated = interpolate_nan(Z, axis=0, max_gap=max_gap)
8
8     return X_interpolated, Y_interpolated, Z_interpolated
```

Quellcode 6.4: Kapselnde 1D-Pipeline: begrenzte Interpolation mit optionaler Glättung

```
1 def smooth_nan_values(x, z, max_nan_values_perc=0.4, max_gap=5,
2                         smoothing=True, window_length=7, polyorder=2):
3     """
4     Interpoliert NaN-Werte in z, aber nur bei kleinerer Lückenanzahl
5     und akzeptablem NaN-Anteil. Optional geglättet mit Savitzky-Golay.
6     """
7     z = np.copy(z)
8     nan_indices = np.isnan(z)
```

```
9     num_nans = np.sum(nan_indices)
10
11    if num_nans / len(z) > max_nan_values_perc:
12        return None
13
14    z = interpolate_limited_nans(z, max_gap=max_gap)
15
16    # Optional glätten
17    if smoothing and z is not None and np.count_nonzero(~np.isnan(z)) > window_length:
18        z = savgol_filter(z, window_length=window_length, polyorder=polyorder)
19
20    return z
```

Quellcode 6.5: Schließen kurzer Täler (Valleys) via begrenzter Interpolation

```
1 def fill_small_valleys(z, window_size=30, depth_threshold=5.0, max_gap=0.5,
2                         smoothing=True, smooth_window=7, polyorder=2):
3     """
4     Glättet kleine Einbrüche ('valleys') in einem Höhenprofil z.
5     """
6     z = np.array(z).copy()
7     rolling_min = pd.Series(z).rolling(window=window_size, center=True,
8                                     min_periods=5).min()
9     diff = rolling_min - z
10
11    # Punkte mit tieferem Einbruch als erlaubt
12    valley_mask = (diff > depth_threshold)
13    z[valley_mask] = np.nan
14
15    z_filled = interpolate_limited_nans(z, max_gap=max_gap)
16
17    # Optional glätten
18    if smoothing and z_filled is not None and np.count_nonzero(~np.isnan(z_filled)) >
19        smooth_window:
20        z_filled = savgol_filter(z_filled, window_length=smooth_window, polyorder=2)
21
22    return z_filled
```

Literaturverzeichnis

- Fischer, Julia; Pochwyt, Kevin (2017): Neuronale Netze. Convolutional Neural Networks am Beispiel eines selbstfahrenden Roboters — Dokumentation. Sphinx-Dokumentation zum Projekt „Autonomer Roboter“. Heinrich-Heine-Universität Düsseldorf. URL: https://user.phil.hhu.de/~petersen/SoSe17_Teampunkt/AR/neuronalenetze.html (Abruf: 28.08.2025).
- Optics and photonics — Lasers and laser-related equipment — Vocabulary and symbols* (2018). International Organization for Standardization. URL: <https://www.iso.org/standard/72944.html> (Abruf: 27.08.2025).
- @lasthash Quick Guide to Surface Roughness Measurement (2016). Enthält Formeln und Definitionen zu Ra und Rz. Mitutoyo Corporation. URL: https://www.mitutoyo.com/webfoo/wp-content/uploads/1984_Surf_Roughness_PG.pdf (Abruf: 28.08.2025).
- Stainless Steel (2025). Angabe von $\sim 49\%$ sichtbarer Reflexion für Edelstahl. A. Zahner Company. URL: <https://www.azahner.com/metal-info/stainless-steel/> (Abruf: 28.08.2025).
- Surface Roughness Parameters — ISO 4287 overview (2025). Erklärt Ra und Rz, Sampling Length X_s und Auswertelänge. KEYENCE. URL: <https://www.keyence.com/ss/products/microscope/roughness/line/parameters.jsp> (Abruf: 28.08.2025).
- Thermal cutting — Classification of thermal cuts — Geometrical product specification and quality tolerances* (2017). International Organization for Standardization. URL: <https://www.iso.org/standard/60321.html> (Abruf: 27.08.2025).