



# 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

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

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>II</b>
<b>Abbildungsverzeichnis</b>	<b>III</b>
<b>Tabellenverzeichnis</b>	<b>IV</b>
<b>Quellcodeverzeichnis</b>	<b>V</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Zielsetzung . . . . .	1
1.2 Vorgehensweise . . . . .	1
<b>2 Grundlagen</b>	<b>3</b>
<b>3 Stand der Technik</b>	<b>4</b>
<b>4 Versuchsplanung zur Identifikation von Schnittabrissgrenzen für Ededlstaibleche</b>	<b>5</b>
<b>5 Datengenerierung mittels experimentellem Plan</b>	<b>7</b>
<b>6 Messmethodik und Datenerfassung in der Messzelle</b>	<b>8</b>
6.1 Anpassung des Handscanner-Setups . . . . .	9
6.2 Optimierung der Vektorberechnung aus 3D-Punktwolken . . . . .	11
6.2.1 Outlier Detection/Correction . . . . .	12
6.2.2 Interpolation . . . . .	14
<b>Literaturverzeichnis</b>	<b>15</b>
<b>Anhang</b>	<b>15</b>

# Abkürzungsverzeichnis

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

<b>CRM</b>	Customer Relationship Management
<b>DHBW</b>	Duale Hochschule Baden-Württemberg
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>ITIL</b>	IT Infrastructure Library
<b>RoI</b>	Return-On-Invest
<b>UCS</b>	Universal Character Set
<b>UTF-8</b>	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 `_dhbw_preamble.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.

---

<sup>1</sup>siehe <http://ctan.org/pkg/acronym>

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Quellcodeverzeichnis

6.1	Werkstoffabhängiges Routing der Handscanner-Setups . . . . .	10
6.2	Outlier Detection/Correction in der Profilverarbeitung (Burr-Kanal) . . . . .	12
6.3	Pipeline-Parameter Outlier Correction (Edelstahl, Burr-Kanal) . . . . .	14
6.4	Pipeline-Parameter Outlier Correction (Baustahl, Burr-Kanal) . . . . .	14
6.1	Interpolation begrenzter Lücken in 1D-Profilen . . . . .	15
6.2	Achswise Interpolation von NaNs in Matrizen . . . . .	16
6.3	Interpolation von X-, Y- und Z-Ebenen . . . . .	16
6.4	Kapselnde 1D-Pipeline: begrenzte Interpolation mit optionaler Glättung . . . . .	16
6.5	Schließen kurzer Täler (Valleys) via begrenzter Interpolation . . . . .	17

# 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. 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. 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 Fokusslage, für verschiedene Edelstahldicken und -güten zu untersuchen. Ziel dieses Testplans ist es, Parameterbereiche zu identifizieren, die zu sogenannten Schnittabrisen führen. Ein Schnittabris 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. 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 Schneidgrate 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. Hierzu werden bestehende Algorithmen zur Bildverarbeitung und Analyse geprüft und entsprechend den spezifischen Anforderungen von Edelstahl optimiert. 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

### 3 Stand der Technik

## 4 Versuchsplanung zur Identifikation von Schnittabrissgrenzen für Ededstahlbleche

Ziel der Versuchsplanung ist die systematische Abgrenzung des prozesssicheren Arbeitsbereichs (Prozessfensters) beim Laserschneiden von Edelstahlblechen sowie die präzise Identifikation der Parameterbereiche, in denen Schnittabrisse auftreten. Unter einem Schnittabritt 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 so gewonnenen Grenzwerte bilden die Grundlage für die spätere Qualitätsmodellierung und dienen der Ableitung belastbarer Empfehlungen für robuste Parametersätze. Zugleich dient die hier beschriebene Vorgehensweise der Ableitung strukturierter Versuchspläne („Experimentalplans“) für die zentrale Versuchsdatenbank.

Die Experimente sind als zweidimensionale Rasterstudien ausgelegt, bei denen jeweils zwei der drei wesentlichen Prozessparameter – Fokusslage, Schnittgeschwindigkeit und Assistgasdruck – variiert werden, während der dritte Parameter konstant gehalten wird. Die Laserleistung wurde in diesen Fällen ebenfalls konstant gehalten. Für jede untersuchte Parameterpaarung wird ein  $3 \times 3$ -Feld gefertigt, in dem eine Größe entlang der horizontalen und die andere entlang der vertikalen Richtung schrittweise 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  $z$  wird auf einem wohldefinierten Referenzniveau  $z_0$  fixiert. Die Studie wird sequenziell für alle drei Kombinationen wiederholt (Fokusslage–Geschwindigkeit bei konstantem Gasdruck, Fokusslage–Gasdruck bei konstanter Geschwindigkeit sowie Geschwindigkeit–Gasdruck bei konstanter Fokusslage), sodass das Prozessfenster in den relevanten Teilräumen konsistent erfasst wird.

Die Wahl der Stufen  $x_i$  und  $y_j$  erfolgt material- und dickenspezifisch. Für jede untersuchte Blechdicke wird zunächst ein plausibler Arbeitsbereich aus praxisüblichen Startwerten und internen Erfahrungswerten abgeleitet und anschließend durch kurze Vorversuche verifiziert. Um Grenzbereiche mit ausreichender Auflösung zu erfassen, wird bei Bedarf adaptiv verfeinert (z. B. durch Intervallhalbierung zwischen zwei Stufen). Die Reihenfolge der Schnitte innerhalb eines Feldes wird randomisiert, um systematische Einflüsse durch fortschreitende Erwärmung, Düsenverschleiß oder driftende Maschinenzustände zu minimieren; die zentrale Kombination  $(x_2, y_2)$  wird zusätzlich repliziert, um Prozess- und Messstreuungen abschätzen zu können.

Die praktische Durchführung jeder  $3 \times 3$ -Matrix beginnt mit einer Funktionskontrolle der Anlage (Düsenabstand, Strahlqualität, Gaszufuhr). Für jede Parameterkombination wird eine definierte Probegeometrie geschnitten. Ein Schnitt gilt als erfolgreich, wenn der Trennschnitt vollständig ist, wenn weder Durchhang noch Wiederaufschmelzen im Schnittspalt beobachtet wird und der Schmelzaustrag kontinuierlich erfolgt. Ein Schnittabritt liegt vor, wenn die Kontur nicht vollständig getrennt ist, der Schnittspalt lokal verschweißt oder ein charakteristischer Abbruch

des Materialaustrags auftritt. Die Beurteilung erfolgt unmittelbar an der Maschine sowie nachgelagert in der Messzelle durch optische Inspektion und Dokumentation der Schnittkante. Die Kriterien werden vorab festgelegt und über alle Versuche konsistent angewendet, um Vergleichbarkeit sicherzustellen.

Sobald ein erster Grenzbereich identifiziert ist, wird das umliegende Parametergebiet gezielt erkundet. Hierzu werden die Stufen in Richtung des beobachteten Grenzverlaufs schrittweise angepasst, 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 empirische Approximation des Prozessfensters. Für jede Kombination aus Blechdicke und Werkstoffgüte entsteht so eine Kontur, die den Bereich prozesssicherer Parameterkombinationen von instabilen Trennbedingungen trennt. Auf Basis dieser Voruntersuchungen werden anschließend detaillierte Versuchspläne („Experimentalplans“) ausgearbeitet. Diese beinhalten systematische Schneidversuche für Edeldstahlbleche mit Dicken bis zu 20 mm; dabei werden bewusst sowohl qualitativ hochwertige als auch minderwertige Schneidergebnisse erzeugt. Ziel ist der Aufbau einer umfassenden, kuratierten Datengrundlage für die Weiterentwicklung des KI-Modells.

Die beschriebene Vorgehensweise verfolgt zwei Ziele: Erstens erlaubt sie die belastbare Identifikation von Schnittabrissgrenzen und damit die sichere Festlegung eines prozessstabilen Arbeitsbereichs für Edeldstahl. Zweitens liefert sie eine qualitativ wie quantitativ aussagekräftige Datenbasis, um das Zusammenspiel der Laserparameter zu verstehen und die Qualität von Schnittkante und Schnittfläche prädiktiv zu bewerten. Damit entsteht ein experimentell abgesichertes Fundament, das sowohl die spätere KI-Modellierung als auch eine reproduzierbare, datengetriebene Parametrierung in der Praxis unterstützt.

## 5 Datengenerierung mittels experimentellem Plan

## 6 Messmethodik und Datenerfassung in der Messzelle

Die Messzelle dient der standardisierten, reproduzierbaren Erfassung aller Messdaten zu den im Rahmen der *Experimentalpläne* geschnittenen Edelstahlbauteilen. Sie ist als sequenzieller Mess-Workflow ausgelegt, in dem ein mehrachsiger KUKA-Industrieroboter die Proben zwischen den Stationen handhabt und so einen kontinuierlichen Materialfluss sowie eine gleichbleibende Prozessführung sicherstellt. 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; die eindeutig ermittelte Proben-ID wird unmittelbar mit den Metadaten aus den *Experimentalplänen* (z. B. Blechdicke, Werkstoffgüte, Soll-Parameter) verknüpft und dient in der Folge als Schlüssel für die Traceability sämtlicher Mess- und Auswertedaten.

Im Anschluss werden an einer bildgebenden Station hochaufgelöste Aufnahmen der ersten Schnittkante erfasst. Hierzu kommt ein Handscanner zum Einsatz, dessen Aufnahmeparameter (Arbeitsabstand, Belichtung, Auflösung) konstant gehalten und protokolliert werden, um eine reproduzierbare Datenbasis zu gewährleisten. Diese Bilddaten bilden die Grundlage für die bildgestützte Qualitätsschätzung des KI-Systems. Ergänzend dazu wird die gleiche Schnittkante mit einem Keyence-System dreidimensional vermessen, sodass eine 3D-Punktwolke des Kantenverlaufs entsteht. Aus dieser Punktwolke werden definierte Profilverläufe abgeleitet und geometrische Kenngrößen berechnet, die der Quantifizierung von Gratbildung und rauheitsnahen Merkmalen dienen. Die so gewonnenen Ist-Kenngrößen fungieren als Referenz für den späteren Abgleich mit der Bildqualitätsschätzung.

Zur vollständigen Dokumentation werden die Schnittkanten zudem mit einer Industriekamera und einem stationären Smartphone-Setup unter definierten Beleuchtungsbedingungen aufgenommen. Die Kombination aus unterschiedlichen Kameras und Beleuchtungen erhöht die Robustheit der visuellen Beurteilbarkeit und unterstützt die spätere manuelle Plausibilisierung von Auffälligkeiten. Sämtliche Rohdaten (Scannerbilder, Industriekamera-/Smartphone-Aufnahmen, 3D-Punktwolken) sowie die daraus abgeleiteten Kenngrößen werden unmittelbar der Proben-ID zugeordnet, qualitätsgeprüft und in die zentrale Datenbank überführt.

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, um kantenbezogene Effekte (z. B. Richtungsabhängigkeiten des Schneidprozesses) erfassen zu können. Abschließend legt der Roboter die vollständig vermessenen Proben an der Endstation geordnet ab. Damit entsteht für jede Probe ein konsistenter, rückverfolgbarer Datensatz aus Identifikation, Rohdaten, abgeleiteten

Kenngrößen und Qualitätsurteilen, der als kuratierte Grundlage für Training und Validierung des KI-Modells dient.

### 6.1 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 (Glanzlichter, lokale Überstrahlungen) und zu einer unzureichenden Abbildung der relevanten Mikrostruktur. In der Folge würden Grate (*Burr*) unterrepräsentiert und die Rauheit (*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 und operativ geführt. Zunächst wird die Belichtungszeit für das Rauheitsbild so eingestellt, dass die Textur der Schnittfläche ohne Sättigung und mit klarer Detailzeichnung sichtbar ist; die Entscheidung erfolgt in dieser Phase bewusst subjektiv, jedoch anhand vorab definierter visueller Kriterien (ausreichender Tonwertumfang, erkennbarer Strukturkontrast, Ausbleiben großflächiger Clipping-Bereiche). 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 ausbrennen. Da die HDR-Komposition durch die Eingangsbilder beeinflusst wird, erfolgt die Abstimmung der HDR-Belichtungen stets nach der Festlegung des Rauheitsbildes.

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 (insbesondere Histogrammverteilung und lokaler Kontrast an der Kantenregion) 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, ohne systematische Verzerrungen einzuführen.

Neben der Belichtung werden alle aufnahmerelevanten Parameter (Arbeitsabstand, Blickwinkel, Fokusslage, Auflösung) konstant gehalten und protokolliert, um die Vergleichbarkeit zwischen Proben und Messserien sicherzustellen. Fehlanpassungen des Setups würden ansonsten zu Fehlklassifikationen führen (z. B. „gut“ statt „unzureichend“) und in der Folge *Burr*- und *Roughness*-Schätzungen verfälschen; dies hätte unmittelbare Konsequenzen für die Optimierung der Schneidparameter. Durch die werkstoffabhängige Justage und die konsequente Dokumentation der Aufnahmeparameter wird die notwendige Reproduzierbarkeit erreicht und die Grundlage für eine belastbare, KI-gestützte Qualitätsschätzung der Edelstahlschnittkanten geschaffen.

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 `Maschinenname-Experimentalplanname-Materialnummer-Bauteildicke-Bauteilnummer`, z. B. `A02280E0005-AiMuWrCjd0-3-050-0176`.

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 6.1 dargestellt und im Messzellenskript implementiert.

Quellcode 6.1: Werkstoffabhängiges Routing der Handscanner-Setups

```
1 public static string NameParserMaterial(string input)
2 {
3     if (string.IsNullOrEmpty(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 }
```

## 6.2 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 Gratomorphologie abgestimmt.

Die nachfolgend beschriebenen Anpassungen betreffen ausschließlich den **Grat-/Burr-Kanal** (Ausreißererkennung und Interpolation). Für die **Rauheitsanalyse (Roughness)** werden die in der Baustahl-Pipeline etablierten Einstellungen *unverändert* übernommen. Interne Funktionsprüfungen unter identischen Aufnahme- und Auswertebedingungen zeigen eine stabile und konsistente Rauheitsbewertung, sodass in diesem Kapitel bewusst auf Abbildungen als Nachweis verzichtet wird.

Bei Edelstahl zeigt sich eine abweichende, ausgeprägt zackige Gratstruktur mit höheren lokalen Gradienten und diskontinuierlichen Profilschnitten. Die bislang implementierte Outlier-Korrektur—konzipiert zur Eliminierung sporadischer Messfehler bei Baustahl—stuft diese hochfrequenten, jedoch materialtypischen Strukturen 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 der Pipeline überarbeitet: (i) die Ausreißererkennung mit nachgelagerter Korrektur und (ii) die Interpolation 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 (z. B. einzelne, isolierte Ausreißerpunkte) und materialtypischen Hochfrequenzanteilen. Entsprechend werden Glättungsschritte zurückgenommen bzw. mit kanten- bzw. strukturerhaltenden Verfahren ausgeführt, 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. Bereiche mit geringer Messsicherheit werden explizit maskiert und in der Auswertung als solche gekennzeichnet, um eine Fehlinterpretation interpolierter Segmente als Messwahrheit zu verhindern. Damit wird erreicht, dass der rekonstruierte 2D-Vektor fehlende Messpunkte plausibel ergänzt, ohne die charakteristische Zackigkeit des Edelstahlgrats zu nivellieren.

### 6.2.1 Outlier Detection/Correction

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 rekonstruiert. In `outlier_correction_burr` wird das Höhenprofil `z_vec` mittels gleitendem Mittelwert (`np.convolve` mit 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 Profilen um, verwendet jedoch standardmäßig einen robusten gleitenden Median (`median=True`) als Referenzsignal (alternativ Mittelwert). 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`). *Hinweis:* Die **Rauheitsanalyse** nutzt unverändert die etablierten Baustahl-Einstellungen; hierfür erfolgt keine Umparametrisierung.

Quellcode 6.2: Outlier Detection/Correction in der Profilvorverarbeitung (Burr-Kanal)

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

```
8
9     outlier_mask = difference > threshold
10    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     Z_series = pd.Series(line)
29     tmp_line = line.copy()
30
31     if median:
32         moving_avg = Z_series.rolling(window_size, min_periods=5,
33         ↪ center=True).median()
34     else:
35         moving_avg = Z_series.rolling(window_size, min_periods=5, center=True).mean()
36
37     difference = np.abs(line - moving_avg)
38     id_outlier = difference > outlier_threshold
39     count_outlier = np.sum(id_outlier)
40
41     if count_outlier / len(line) > max_outlier_values_perc:
42         return None, None
43
44     tmp_line[id_outlier] = np.nan
45     tmp_line = interpolate_limited_nans(tmp_line, max_gap=max_gap)
46
47     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 6.3 und Quellcode 6.4). Für die **Rauheitsanalyse** wurden *keine* Parameteränderungen vorgenommen.

Quellcode 6.3: 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 6.4: 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]
```

### 6.2.2 Interpolation

Die Interpolation dient der Rekonstruktion fehlender Messwerte (NaN), die durch Ausreißermaskierung oder unvollständige Erfassung entstehen. Ziel ist eine *konservative* Wiederherstellung: Es werden nur kurze, lokal begrenzte Lücken gefüllt; größere Ausfälle bleiben als Unsicherheitsbereiche markiert und fließen nicht in abgeleitete Kenngrößen ein. Der Rekonstruktionsschritt nutzt ausschließlich lokale Nachbarschaftsinformationen, um die charakteristische Profil- bzw. Kantenstruktur zu erhalten und unnötige Verfälschungen zu vermeiden. *Die hier beschriebenen Rekonstruktionsschritte werden im Rahmen der Edelstahl-Anpassungen primär für den Burr-Kanal genutzt; die Rauheits-Pipeline verbleibt mit den bewährten Baustahlparametern.*

Das Verfahren wird sowohl auf eindimensionale Profile (entlang der Schnittkante) als auch auf flächige Daten angewendet. Bei Flächen erfolgt die Rekonstruktion achsweise entlang sinnvoller Richtungen, sodass die lokale Geometrie bestmöglich bewahrt bleibt. Zusätzlich werden rekonstruierte Punkte explizit gekennzeichnet, sodass die Nachvollziehbarkeit zwischen gemessenen und interpolierten Werten jederzeit gegeben ist. Plausibilitätsprüfungen (u. a. Anteil fehlender Werte, maximale Lückenlänge) stellen sicher, dass nur hinreichend kleine und stabile Lücken geschlossen werden; andernfalls wird auf eine Rekonstruktion verzichtet.

In der Interpolations-Pipeline wurden *keine Konstanten geändert*. Insbesondere bleiben Grenzwerte für die maximale Lückenlänge sowie etwaige Glättungs- und Fensterparameter gegenüber der Baustahlkonfiguration unverändert, um die Vergleichbarkeit zwischen Materialklassen sicherzustellen. Der zugehörige Quellcode ist im Anhang dokumentiert.

# 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     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     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 kleiner 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
```