

Entwicklung eines CAM-Systems für 2D und 2.5D Laserbearbeitung

Johannes Gysel

Bachelorarbeit

FS 2015

Prof. Dr. Konrad Wegener

Maximilian Warhanek

Josquin Pfaff

Zusammenfassung

Die Mikrobearbeitung mittels Ultrakurz-Puls-Lasern erfreut sich zunehmender Beliebtheit. Durch die steigende Komplexität der Bearbeitungsmaschinen steigen auch die Anforderungen bei der Fertigung. Um diese realisieren zu können, werden CAM Programme verwendet.

Bestehende CAM Programme werden den Anforderungen nicht immer gerecht. Deshalb wurde im Rahmen dieser Arbeit ein neues CAM Programm für die 2.5D Laserbearbeitung entwickelt.

Dieses Programm kann aus den dreidimensionalen Geometriedaten (Stl-Datei) des gewünschten Werkstücks die Bahnlinien zu dessen Fertigung berechnen. Diese können als NC-Code für diverse Maschinen abgespeichert und ausgeführt werden.

Es wird beschrieben wie das Programm aufgebaut ist und wie die einzelnen Teilschritte und Funktionen implementiert wurden.

Bei der 2.5D Laserbearbeitung können diverse Geometriefehler am Werkstück auftreten. Durch innovative Methoden bei der Berechnung der Bahnlinien können diese reduziert oder komplett vermieden werden. Wie diese Lösungsansätze aussehen und wie sie im Programm realisiert wurden ist beschrieben.

Zur einfacheren Bedienung des Programms wurde eine Benutzeroberfläche erstellt. Über diese können die Parameter zur Laserbearbeitung bequem eingegeben werden. Die berechneten Laserbahnen sind direkt über eine Grafik einsehbar. Dadurch können schon vor der Bearbeitung Fehler vermieden werden.

Der durch das CAM Programm berechnete NC-Code konnte von der Maschine korrekt interpretiert und ausgeführt werden. In diversen Versuchen wurden so die gewünschten Werkstücke gefertigt. Durch die Wahl geeigneter Prozessparameter wurde das Material nicht nur abgetragen sondern auch die Oberflächenqualität gesteigert.

Dank dem modularen Aufbau des CAM Programms kann dieses einfach an weitere Bedürfnisse und Prozesse angepasst werden. Dafür steht der Programmcode jedem zur Verfügung.

Abstract

Micromachining with ultrashort pulsed lasers is becoming increasingly popular. Due to the rising complexity of the processing machines there is also a rising complexity in production. Therefore CAM programs are used to help in the production process.

It is not easy to find CAM programs that reach all requirements. Therefor a new CAM program for 2.5D laser processing was developed in this thesis.

This program can calculate the trajectories for production from the three-dimensional geometry (stl-file) of the desired workpiece. These trajectories can be saved as g-code, which can be adjusted for different laser processing machines.

It is described how the program is structured and how the individual steps and functions are implemented.

In 2.5D laser processing a number of geometric errors may occur. They can be reduced or completely avoided through innovative methods in the calculation of the trajectories. How these solutions look like and how they are implemented is described.

A graphical user interface was created. Thus, the parameters for laser processing can be entered more easily. The calculated trajectories can be viewed on a graphic. Therefor errors can be avoided even before production.

The G-code generated by the CAM program was correctly executed by the machine. In various experiments the desired workpieces were produced. By choosing suitable process parameters, the material was not only removed there was also an increase in surface quality.

Due to the modular structure of the CAM program, it is easy to adapt it for other needs or processes. For these purposes the program code is available for everyone

Entwicklung eines CAM-Systems für 2D und 2.5D Laserbearbeitung

Bachelor/Semester/Masterarbeit

Problemstellung

Der Einsatz von Ultrakurz-Puls-Lasertechnik in der Mikrobearbeitung erfreut sich steigendem Interesse in diversen Industrien (Werkzeug, Medizin, Luft und Raumfahrt, etc.). Neueste Steuerungssysteme erlauben die synchrone Anwendung von optischen Achsen und konventionellen CNC-Achsen (Abb. 1). Die IWF-Lasergruppe befasst sich intensiv mit der Entwicklung von Laserbearbeitungsverfahren, welche diese Kombination von Achsen nutzen, um den speziellen Anforderungen der Mikrobearbeitung gerecht zu werden. Zur industriellen Umsetzung dieser Prozesse sind entsprechende CAM-Systeme notwendig.

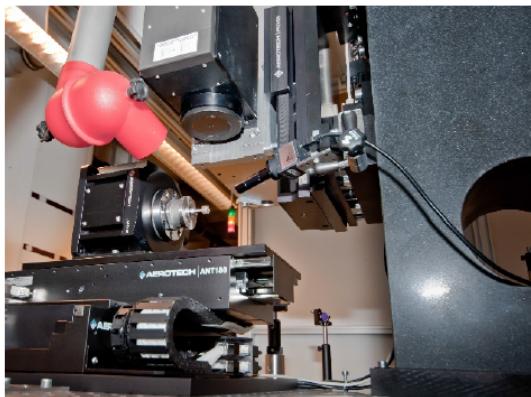


Abb. 1: Scankopf – CNC Kombination [Aerotech]

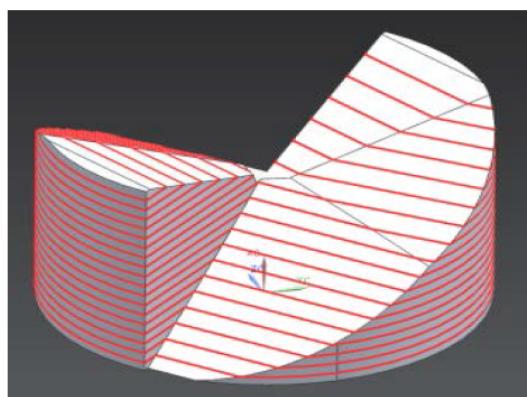


Abb. 2: Slicing

Aufgaben

Diese Arbeit stellt die Basis für ein umfassendes CAM-System bestehend aus mehreren Modulen dar. Es sollen Algorithmen für die 2D-Bearbeitung (Schraffuren), 2.5D-Bearbeitung (Slicing, Abb. 2) und eine grafischen Nutzeroberfläche programmiert werden.

- Literaturrecherche
- Programmierung: 2D Bearbeitung
- Programmierung: Slicing für eine 2.5D Bearbeitung
- Programmierung: Graphische Nutzeroberfläche
- Testen der entwickelten Programme

Die Aufgaben können im Falle von Bachelor- und Semesterarbeiten zwischen mehreren Studenten aufgeteilt werden.

Kontakt

Maximilian Warhanek	PFA H43	044 633 78 40	warhanek@iwf.mavt.ethz.ch
Josquin Pfaff	PFA H43	044 633 62 54	pfaff@iwf.mavt.ethz.ch
Josef Stirnimann	CLA G7	044 632 31 73	stirnimann@inspire.ethz.ch

Danksagung

An dieser Stelle bedanke ich mich herzlich bei meinen Betreuern Maximilian Warhaneck und Josquin Pfaff. Sie haben mir den Freiraum gelassen, um dieses Projekt mit eigenen Vorstellungen bewerkstelligen zu können. Durch ihr kompetentes Fachwissen haben sie mich frühzeitig auf Probleme aufmerksam gemacht und mir bei der Realisierung geholfen.

Ein weiterer Dank geht an Herrn Termoellen von der Aerotech GmbH. Dank seinem guten Hinweis zur Steuerungssoftware blieb mir viel Mühe und Arbeit erspart.

Zudem möchte ich mich herzlich bei David Gonon bedanken. Er hat mit seiner Programmierung eine sehr gute Vorarbeit geleistet. Ohne seine Unterstützung hätte ich die Arbeit in diesem Umfang nicht rechtzeitig fertigstellen können.

Inhaltsverzeichnis

Zusammenfassung	i
Abstract	iii
Danksagung	vii
Inhaltsverzeichnis	ix
Abbildungsverzeichnis	xiii
Tabellenverzeichnis	xv
Nomenklatur	xvii
1. Einleitung	1
2. Stand der Technik	3
2.1 Ultrakurz-Puls-Lasertechnik	3
2.2 Mikrobearbeitung durch Ultrakurz-Puls-Laser	3
2.2.1 2.5D Laserbearbeitung	4
2.3 Scankopf	4
2.4 CAM	5
2.4.1 SAMLight	5
2.4.2 Cura	6
2.5 Konstruktion der Modellgeometrie	7
2.6 Das Stl-Format	8
2.7 Slicing	9
2.8 Schraffuren	9
2.9 Skywritelinien	10
2.9.1 Problemstellung	10
2.9.2 Lösungsansatz	10
2.9.3 Abschätzung der Skywritelängen	11

2.10	Die Rillenbildung.....	12
2.11	Strahlkompensation.....	12
2.11.1	Problemstellung	12
2.11.2	Lösungsansatz	13
2.11.3	Der Konturabstand bei der Strahlkompensation	14
2.11.4	Das Problem der Strahlkompensation.....	14
2.11.5	Die Richtung in welche die Kanten verschoben werden.....	14
2.12	Straight Skeleton.....	15
2.13	Konturumrandung	16
2.13.1	Der Breakangle	16
2.13.2	Strahlkompensation bei der Konturumrandung	16
3.	Versuchsaufbau und Hilfsmittel.....	19
3.1	Strahlquelle	19
3.2	Der Strahlengang	20
3.3	Die mechanischen und optischen Achsen.....	21
3.4	CNC Steuerung	21
3.4.1	Übersicht des verwendeten NC-Codes.....	22
3.4.2	Übersetzung der einzelnen Bahnlinien in NC-Code	22
3.4.3	NC-Befehle zu Beginn des Codes.....	22
3.4.4	Ende des NC-Codes	22
3.4.5	Ausführen von langem NC-Code.....	23
3.5	Matlab	23
3.5.1	Stl-Datei in Matlab importieren	23
3.6	Oberflächenrauheit.....	24
3.7	Optische Messgeräte	24
4.	CAM Programmierung	25
4.1	Übersicht	25
4.2	Wahl der Schnitthöhen.....	26
4.3	Berechnung der Schnittkonturen.....	26
4.3.1	Berechnung der Schnittpunkte	27
4.3.2	Verbinden der Schnittpunkte zu geschlossenen Konturen	29
4.4	Berechnung der Schraffuren	29
4.4.1	Spezialfälle bei der Schraffurberechnung	30
4.4.2	Verbinden der Schnittpunkte zu Schraffuren	31
4.5	Implementierung Skywrite.....	32
4.6	Implementierung variabler Schraffurwinkel.....	32
4.7	Strahlkompensation.....	33
4.7.1	Berechnung der Winkelhalbierenden.....	33
4.7.2	Statusberechnung der Eckpunkte	34

4.7.3	Eckpunkte mit Status 0	35
4.7.4	Berechnung des Straight Skeletons bei Konturen mit konvexen Eckpunkten.....	35
4.7.5	Berechnung des kritischen Punktes bei einem konvexen Eckpunkt.....	36
4.7.6	Berechnung der neuen Kontur bei einem Edge Event.....	37
4.7.7	Straight Skeleton Berechnung bei Konturen mit konkaven Eckpunkten	37
4.7.8	Berechnung des kritischen Punktes bei einem konkaven Eckpunkt.....	38
4.7.9	Merge Event	39
4.7.10	Korrektur von Konturen mit Eckpunkten vom Status 2	40
4.7.11	Speicherung einer geschlossenen Konturen	40
4.7.12	Implementation des eigenen Straight Skeleton Algorithmus	41
4.7.13	Nachteil der Strahlkompensation.....	43
4.8	Benutzeroberfläche.....	43
4.8.1	Aufbau der Benutzeroberfläche.....	43
4.8.2	Parametereingabe zur Bahnberechnung	44
4.8.3	Erstellen des NC-Codes.....	45
4.8.4	Einstellungsmöglichkeiten zur grafischen Darstellung	46
5.	Ergebnisse und Diskussion.....	47
5.1	Werkstück mit ETH Logo	47
5.1.1	Die gewählten Bearbeitungsparameter für das ETH Logo.....	48
5.1.2	Der Materialabtrag beim ETH Logo	48
5.2	Werkstück mit abgetragenem Laserwarnzeichen	49
5.2.1	Die Bearbeitungsparameter für das Laserwarnzeichen	50
5.2.2	Oberflächenqualität beim Laserwarnzeichen	50
6.	Schlussfolgerung und Ausblick	53
Anhang	55
A.1	Anleitung zum Ausführen des NC-Codes	55
A.2	Daten auf der beigelegten CD	57
A.2.1	Sourcecode CAM Programm.....	57
A.2.2	Kompiliertes CAM Programm	57
A.2.3	Stl-Dateien.....	57
A.2.4	NC-Code.....	57
A.2.5	Grafiken.....	57
Literaturverzeichnis	59
Plagiatsklärung	61

Abbildungsverzeichnis

Abbildung 2.1:	Leistungsabgabe eines Ultrakurz-Puls-Lasers	3
Abbildung 2.2:	2.5D Laserbearbeitung.....	4
Abbildung 2.3:	Aufbau des Scankopfs	5
Abbildung 2.4:	Prozesskette mit CAM.....	5
Abbildung 2.5:	Benutzeroberfläche SAMLight.....	6
Abbildung 2.6:	Benutzeroberfläche von Cura	7
Abbildung 2.7:	Geometrie des Werkstücks und des abgetragenen Volumens	7
Abbildung 2.8:	Stl-Format mit Normalenvektoren.....	8
Abbildung 2.9:	Beim Slicing entstehen zweidimensionale Schnittkonturen.....	9
Abbildung 2.10:	Die Schraffuren werden aus den geschlossenen Konturen berechnet	9
Abbildung 2.11:	Laserbearbeitung ohne Skywrite	10
Abbildung 2.12:	Ohne Skywrite resultiert ein erhöhter Materialabtrag entlang der Kontur	10
Abbildung 2.13:	Mit Skywrite ist der Materialabtrag gleichmässig.....	11
Abbildung 2.14:	Laserbearbeitung mit Skywrite.....	11
Abbildung 2.15:	Rillenbildung	12
Abbildung 2.16:	Variabler Schraffurwinkel	12
Abbildung 2.17:	Laserbearbeitung ohne Strahlkompensation.....	13
Abbildung 2.18:	Berechnung einer neuen, kleineren Kontur	13
Abbildung 2.19:	Laserbearbeitung mit Strahlkompensation	14
Abbildung 2.20:	Bei der Strahlkompensation können unerwartete Konturen entstehen	14
Abbildung 2.21:	Verschiebung der Kanten entgegen der Normalenvektoren	15
Abbildung 2.22:	Veranschaulichung eines Straight Skeleton.....	15
Abbildung 2.23:	Konturumrandung links ohne, rechts mit Skywritelinien.....	16
Abbildung 2.24:	Skywritelinien werden erst nach Überschreitung des Breakangles eingefügt	16
Abbildung 2.25:	Strahlkompensation bei der Konturumrandung.....	17
Abbildung 3.1:	Der verwendete Ultrakurz-Puls-Laser	19
Abbildung 3.2:	Optische Instrumente im Strahlengang.....	20
Abbildung 3.3:	Der Scankopf, die mechanische Z-Achse und der Werkzeugtisch	21
Abbildung 3.4:	Nachträgliche Berechnung des Normalenvektors	24
Abbildung 4.1:	Vereinfachte Übersicht der CAM Programmierung.....	25
Abbildung 4.2:	Der erste Schnitt wird direkt auf der Oberfläche angesetzt	26
Abbildung 4.3:	Der Materialabtrag kommt der Sollgeometrie am nächsten	26
Abbildung 4.4:	Darstellung der Schnittgeraden zwischen Dreieck und Schnittebenen	27

Abbildung 4.5:	Mehrere Eckpunkte liegen genau in einer Schnittebene	28
Abbildung 4.6:	Die einzelnen Schnittpunkte werden zu Schnittkonturen verbunden.....	29
Abbildung 4.7:	Berechnung der Schnittpunkte durch das vertikale Linienraster.....	30
Abbildung 4.8:	Spezialfälle, die bei der Schraffurberechnung auftreten können	30
Abbildung 4.9:	Die Schnittpunkte können auf verschiedene Weise verbunden werden.....	31
Abbildung 4.10:	Berechnung der Skywritelinien.....	32
Abbildung 4.11:	Berechnung der Schraffurbilder mit geneigtem Bahnlinien	33
Abbildung 4.12:	Berechnung der Winkelhalbierenden.....	34
Abbildung 4.13:	Berechnung des Status für Eckpunkt E1	35
Abbildung 4.14:	Bei einem Edge Event verschmelzen zwei Eckpunkte zu einem neuen	36
Abbildung 4.15:	Bestimmung des kritischen Punktes bei konvexen Eckpunkten	36
Abbildung 4.16:	Berechnung der Position des aufgerückten neuen Eckpunktes	37
Abbildung 4.17:	Bei einem Split Event entstehen zwei neue Teilkonturen.....	38
Abbildung 4.18:	Berechnung des kritischen Punktes bei einem konkaven Eckpunkt	39
Abbildung 4.19:	Bei einem Merge Event verschmilzt die äussere Kontur mit der inneren.....	39
Abbildung 4.20:	Eckpunkte mit Status 2 müssen speziell behandelt werden	40
Abbildung 4.21:	Benutzeroberfläche des CAM Programms.....	44
Abbildung 4.22:	Anpassung der NC-Befehle.....	45
Abbildung 4.23:	Grafische Darstellung der Stl-Datei, Konturen und Laserbahnen.....	46
Abbildung 5.1:	Mikroskopaufnahme des ETH Logos	47
Abbildung 5.2:	Detailaufnahmen des abgetragenen ETH Logos	48
Abbildung 5.3:	Mikroskopaufnahme des gefertigten Laserwarnzeichens	49
Abbildung 5.4:	Detailaufnahmen des Laserwarnzeichens	50
Abbildung A.1:	Benutzeroberfläche des Motion Composer von Aerotech	55

Tabellenverzeichnis

Tabelle 3.1:	Spezifikationen des DUETTO Lasers [13].....	20
Tabelle 3.2:	Die wichtigsten NC-Befehle für die CNC-Steuerung A3200 von Aerotech	22
Tabelle 3.3:	Vorbereitung des NC-Codes.....	22
Tabelle 3.4:	Ende des NC-Codes.....	23
Tabelle 5.1:	Verwendete Parameter für das Fertigen des ETH Logos	48
Tabelle 5.2:	Verwendete Parameter zur Fertigung des Laserwarnzeichens	50

Nomenklatur

P_a	Durchschnittliche Laserleistung	[W]
P_m	Maximale Laserleistung	[W]
f_P	Pulsfrequenz des Lasers	[kHz]
E_p	Pulsenergie	[μ J]
τ_p	Pulsdauer	[ps]
M^2	Beugungsmasszahl	[$-$]
λ	Wellenlänge	[nm]
d_f	Strahldurchmesser im Fokus	[μ m]
d_s	Strahldurchmesser vor dem Scankopf	[μ m]
f	Brennweite	[mm]
z_R	Rayleighlänge	[μ m]
v_s	Scangeschwindigkeit	[mm/s]
s	Beschleunigungsstrecke	[μ m]
l	Skywritelänge	[μ m]
a_b	Linienabstand zwischen den Laserbahnen	[μ m]
a_z	Schichtdicke	[μ m]
Ra	Arithmetischer Mittelwert der Profilordinate	[μ m]
α	Änderungswinkel zweier aufeinander folgenden Bahnlinien	[Grad]
β	Breakangle	[Grad]
γ	Winkel zwischen zwei Richtungsvektoren	[Grad]
δ	Variabler Schraffurwinkel	[Grad]
\vec{v}	Richtungsvektor	

\vec{n}	Normierter Richtungsvektor
k	Kreuzprodukt
E	Eckpunkt
S	Schnittpunkt
K	Kritischer Punkt
A	Abstand
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
RTC	Real Time Controller

1

Einleitung

Im Jahr 1808 hatte Joseph-Marie Jacquard die grandiose Idee, einen Webstuhl zu konstruieren, der durch ein Lochkartensystem gesteuert wurde. Auf diesen Lochkarten waren allerlei Informationen über das zu webende Muster enthalten. Dieser Webstuhl war die erste programmierbare Maschine und damit der erste Schritt in der Automatisierung der Weberei [1].

Rund 200 Jahre nach Joseph-Marie Jacquards Erfindung hat sich in der Maschinenindustrie an diesem Grundprinzip nicht viel geändert. Immer noch sind es Schritt für Schritt ausgeführte Befehle, die den Maschinen zum Verrichten komplexer Tätigkeiten helfen. Heutzutage werden diese Befehle nicht mehr auf Lochkarten, sondern als binäre Daten auf dem Computer abgespeichert.

Durch die steigende Komplexität der Produktionsmaschinen und der gefertigten Produkte steigt auch die Komplexität zur Erstellung dieser Befehle. Waren es früher Menschen, die einzelne Löcher in Lochstreifen stanzten, sind es heute Computerprogramme, welche diese Befehle berechnen und aneinander reihen.

Durch die Verwendung von Lasern in der Fertigungstechnik sind neue Verfahren bei der Herstellung entstanden. Gerade in der Mikrobearbeitung ergeben sich durch den Einsatz von gepulsten Lasern neue Möglichkeiten.

In diesem Bereich ist man durch die Komplexität der Bearbeitung auf Computerprogramme angewiesen.

Auf dem Markt gibt es eine Vielzahl solcher Programme. Diese werden den eigenen Anforderungen nicht immer gerecht. Ein eigenes Programm kann beliebig an seine Bedürfnisse angepasst werden. Deshalb wurde im Rahmen dieser Arbeit ein solches entwickelt.

2

Stand der Technik

2.1 Ultrakurz-Puls-Lasertechnik

Bei einem Ultrakurz-Puls-Laser wird die Energie im Resonator kontinuierlich aufgebaut und anschliessend durch die Erzeugung eines sehr kurzen, intensiven Lichtimpulses abgegeben [2]. Die Dauer eines Impulses ist nur wenige Pikosekunden (siehe Abbildung 2.1). Obwohl die durchschnittliche Laserleistung P_a nur wenige Watt beträgt, kann die maximale Leistung der Lichtimpulse P_m mehrere Gigawatt erreichen. Wird diese Energie auf eine sehr kleine Fläche fokussiert, sind die erreichbaren Intensitäten enorm. Dadurch lassen sich sehr harte Materialien wie Hartmetall oder sogar Diamant bearbeiten. Die Bearbeitungszeit durch die einzelnen Pulse ist so kurz, das der thermische Einfluss auf das Werkstück gering ist.

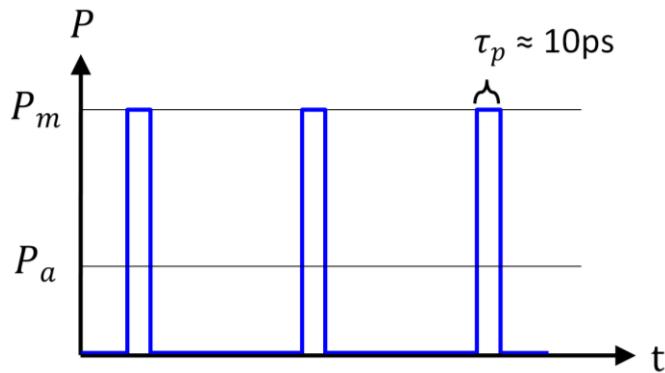


Abbildung 2.1: Leistungsabgabe eines Ultrakurz-Puls-Lasers

2.2 Mikrobearbeitung durch Ultrakurz-Puls-Laser

Bei der Mikrobearbeitung durch Ultrakurz-Puls-Laser wird das Material durch viele kleine Lichtimpulse des Lasers abgetragen. Dadurch lassen sich feine und komplexe Strukturen in harten Materialien realisieren. In einem Artikel von Gregory Eberle und Claus Dold [3] ist die Fertigung

und Anwendung solch feiner Strukturen geschildert. Gerade bei der Herstellung von Wendeschneidplatten für die spanende Bearbeitung bieten sich neue Möglichkeiten. Herkömmliche Schleifprozesse stoßen hier an ihre Grenzen. In der Bachelorarbeit von Lukas Lehner [4] wurden mittels Laserbearbeitung Wendeschneidplatten für den Drehprozess gefertigt. Die Spanleitstufen wurden mittels 2.5D Laserbearbeitung abgetragen.

2.2.1 2.5D Laserbearbeitung

Bei der 2.5D Laserbearbeitung fährt man mit dem Laserstrahl in engen Bahnen über das Werkstück. Dabei liegt der Fokus des Strahls genau auf der Werkstückoberfläche. Der Laser sendet in regelmässigen Abständen kurze, intensive Lichtimpulse aus. Diese Lichtimpulse erwärmen das Material im Fokus so stark, dass es lokal schmilzt und verdampft. Dadurch ist ein gezielter Materialabtrag möglich. Dieser Vorgang kann Schicht für Schicht wiederholt werden, bis die gewünschte Zielgeometrie erreicht ist. In Abbildung 2.2 ist dieser Prozess veranschaulicht.

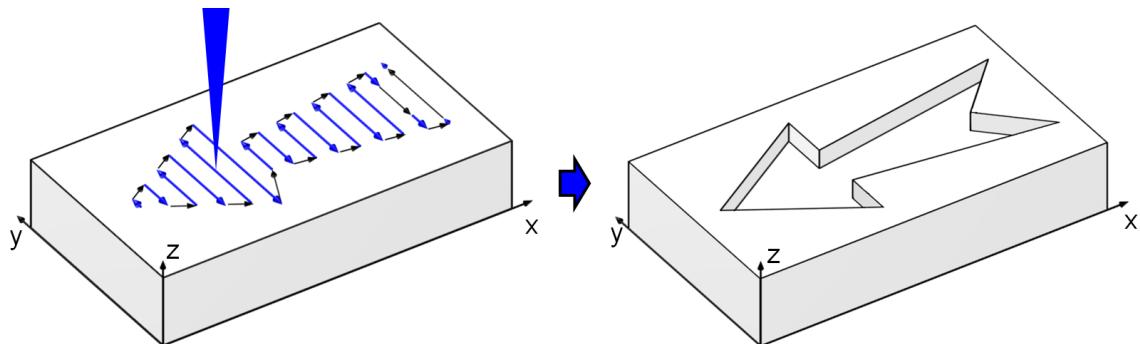


Abbildung 2.2: 2.5D Laserbearbeitung

Für die 2.5D Laserbearbeitung benötigt man drei Bearbeitungsachsen. Die Z-Achse ist die Zustellachse und wird auch Halbachse genannt. Mit dieser wird zwischen den abgetragenen Schichten der Fokus neu justiert. Mit den X- und Y-Achsen, kann der Laserstrahl innerhalb der Ebene positioniert werden.

2.3 Scankopf

Mit dem Scankopf kann der Laserstrahl mithilfe zweier Spiegel an die richtige Zielposition umgelenkt werden [5]. Dafür gibt es einen Spiegel und einen Galvomotor, der den Laserstrahl in die X-Richtung umlenkt und einen Spiegel und einen Galvomotor, der den Laserstrahl in die Y-Richtung umlenkt. Eine F-theta Linse sorgt dafür, dass der Fokus des Laserstrahls immer auf einer Z-Ebene liegt, egal unter welchem Winkel dieser eintrifft. In Abbildung 2.3 ist der schematische Aufbau des Scankopfs dargestellt.

Die Galvomotoren sind dynamisch und ermöglichen ein schnelles Kippen der einzelnen Spiegel. Dadurch erreicht der Laserstrahl auf der Oberfläche hohe Scangeschwindigkeiten von mehreren Metern pro Sekunde.

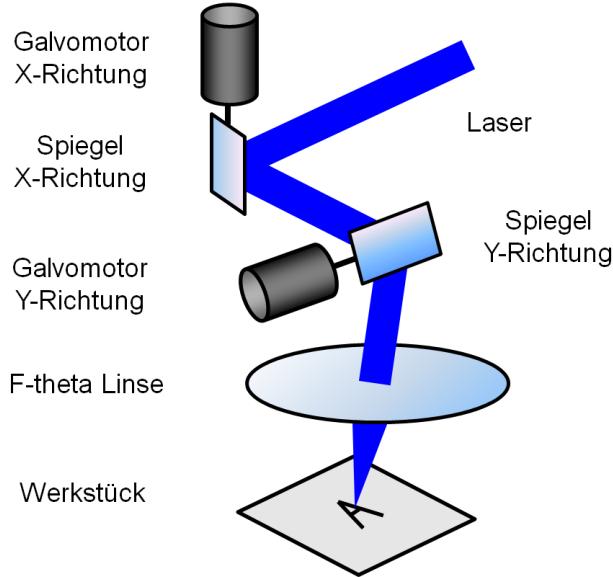


Abbildung 2.3: Aufbau des Scankopfs

2.4 CAM

CAM steht für Computer Aided Manufacturing, ein Programm also, das einem bei dem Herstellungsprozess des Werkstücks unterstützt. In Abbildung 2.4 ist die Prozesskette, in der sich das CAM befindet dargestellt. Am Anfang dieser Kette ist das CAD Programm. CAD steht für Computer Aided Design. Mit diesem wird die dreidimensionale Geometrie gezeichnet. Das geometrische Modell übergibt man dem CAM. Dieses berechnet daraus die einzelnen Laserbahnen, die für die Laserbearbeitung nötig sind. Aus diesen Berechnungen wird der NC-Code erstellt. Dieser kann von der Maschine interpretiert und ausgeführt werden, um das Werkstück herzustellen.

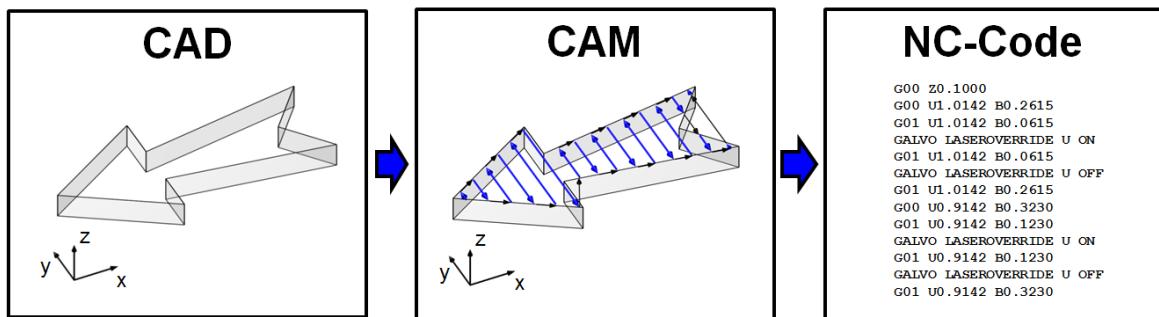


Abbildung 2.4: Prozesskette mit CAM

Um eine Vorstellung zu bekommen, wie ein CAM Programm aufgebaut ist, wurden bestehende Programme getestet.

2.4.1 SAMLight

SAMLight ist eine kommerzielle Software, die von SCAPS GmbH vertrieben wird. Es ist für die Verwendung und Ansteuerung von Scanköpfen ausgelegt [6].

In SAMLight können diverse zweidimensionale oder dreidimensionale Geometriedaten importiert werden. Im Programm können über die Eingabe verschiedener Parameter die Bahnlinien für die Laserbearbeitung berechnet werden. Ist das Programm über eine RTC-Karte mit dem Scankopf verbunden, können die berechneten Laserbahnen abgefahrt werden.

SAMLight bietet die Möglichkeit Stl-Dateien zu importieren und durch ein Slicing Verfahren zweidimensionale Schichten zu berechnen. Diese Schichten können mit dem Laser nacheinander abgetragen werden.

Da die meisten Scanköpfe nur zwei optische Achsen haben, ist eine 2.5D Bearbeitung nur mithilfe einer zusätzlichen Achse möglich. Dazu kann man einen Fokusshifter benutzen. Dieser kann gemeinsam über die RTC-Karte mit dem Scankopf über SAMLight angesteuert werden. Mit SAMLight gibt es aber keine Möglichkeit den Fokus über mechanische Achsen einzustellen.

In Abbildung 2.5 ist ein Screenshot der Benutzeroberfläche von SAMLight ersichtlich. Die berechneten Laserbahnen der obersten Schicht sind dabei rosa dargestellt.

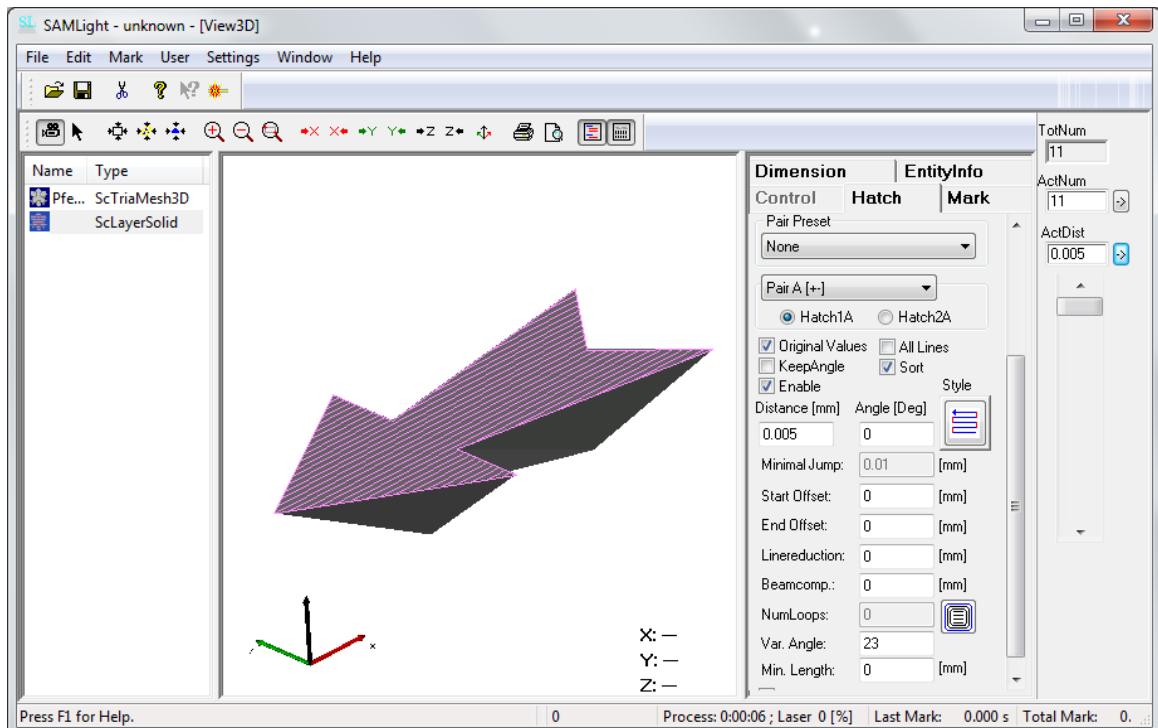


Abbildung 2.5: Benutzeroberfläche SAMLight

2.4.2 Cura

Cura ist ein open source Programm, das zur Benutzung des 3D-Druckers namens Ultimaker benötigt wird [7]. Bei diesem 3D-Drucker wird erwärmter Kunststoff durch eine Düse ausgestossen. Die Düse wird durch drei computergesteuerte Achsen entlang berechneter Bahnlinien bewegt. Der ausgestossene Kunststoff kühlt ab und verhärtet sich. Durch mehrere aufgetragene Kunststoffbahnen kann das gewünschte Werkstück Schicht für Schicht hergestellt werden.

Die Benutzeroberfläche von Cura ist übersichtlich und einfach zu bedienen. Wie bei SAM Light können die Schichten zur besseren Darstellung einzeln angezeigt werden. In Abbildung 2.8 ist ein Screenshot von Cura ersichtlich. Die berechneten Kunststoffbahnen, die beim 3D-Druck aufgetragen werden, sind rot, grün und gelb dargestellt.

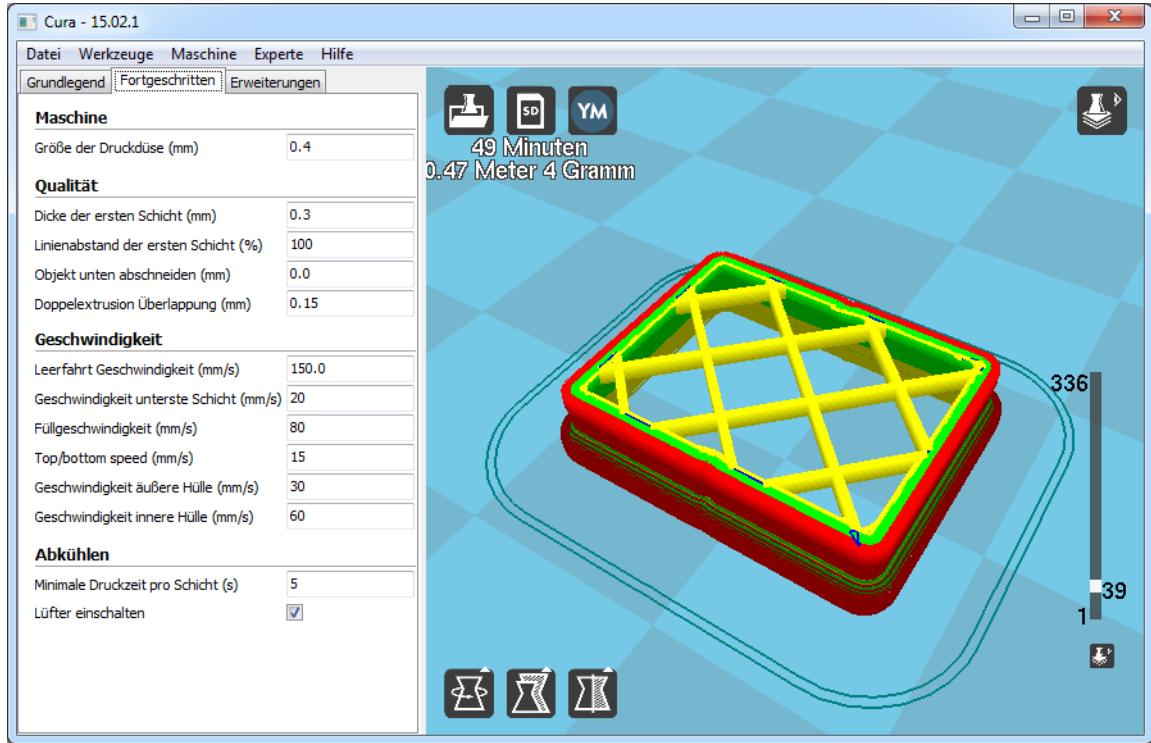


Abbildung 2.6: Benutzeroberfläche von Cura

Das 3D-Drucken ist der 2.5D Laserbearbeitung ähnlich. Bei beiden Verfahren wird das Werkstück durch mehrere Schichten hergestellt. Bei der 2.5D Laserbearbeitung wird das Material in einzelne Schichten abgetragen. Beim 3D-Drucken wird das Material in einzelnen Schichten aufgetragen.

2.5 Konstruktion der Modellgeometrie

In einem CAD Programm wird die Geometrie des Volumens, das später durch den Laser abgetragen wird, gezeichnet (siehe Abbildung 2.7 rechts). Diese Geometrie entspricht genau dem Negativ des Werkstücks, das später durch die Laserbearbeitung hergestellt wird (siehe Abbildung 2.7 links).

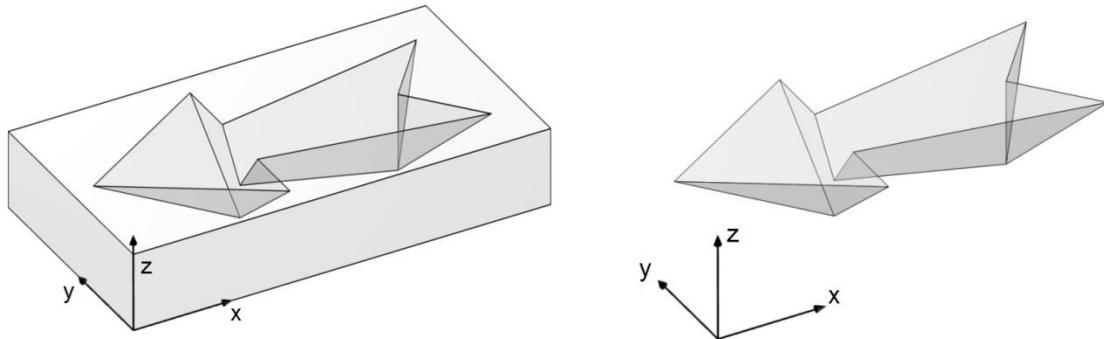


Abbildung 2.7: Geometrie des Werkstücks und des abgetragenen Volumens

Es steht einem frei, mit welchem Programm die Erstellung dieser Geometrie geschieht. Entscheidend ist letztlich nur, dass die gezeichnete Geometrie von dem CAD Programm im binären Stl-Format abgespeichert werden kann. Das Stl-Format wird von den meisten dreidimensionalen Bearbeitungs-Programmen unterstützt [8].

Zu erwähnen ist, dass die Geometriedaten der Stl-Datei am Koordinatenursprung des CAD Programms ausgerichtet sind. Die Koordinaten der Laserbahnen, die später durch diese Geometrie berechnet werden, richten sich also auch nach diesem Koordinatenursprung. Deshalb ist darauf zu achten, dass der Koordinatenursprung im CAD Programm derselbe ist wie der des Werkstücks auf der Maschine.

Idealerweise ist die XY-Ebene auf der Oberfläche des Werkstücks zu wählen. Dadurch steht die positive Z-Achse senkrecht auf der Oberfläche des Werkstücks und der Nullpunkt der Z-Achse ist genau auf der Oberfläche des Werkstücks.

Diese Empfehlung ist nicht zwingend einzuhalten. Entscheidend ist nur, dass die Z-Achse bei der Modellgeometrie richtig orientiert ist. Sie entspricht bei der 2.5D Laserbearbeitung der Zustellachse.

2.6 Das Stl-Format

Ein Computer kennt keine wirklichen Körpervolumen. Das einzige womit ein Computer umgehen kann, sind Zahlen. Damit der Computer dreidimensionale Objekte interpretieren kann müssen diese durch einzelne Zahlen beschrieben werden.

Das Stl-Format ist ein gängiges Format um dreidimensionale Geometrien zu speichern. Dabei wird die Oberfläche des dreidimensionalen Körpers durch ein geschlossenes Netz von Dreiecken repräsentiert. Von jedem Dreieck wird die Position der drei Eckpunkte gespeichert, also eine X-, Y-, und Z-Koordinate für jeden Eckpunkt [8]. In Abbildung 2.8 sind die Eckpunkte durch kleine, schwarze Kugeln dargestellt.

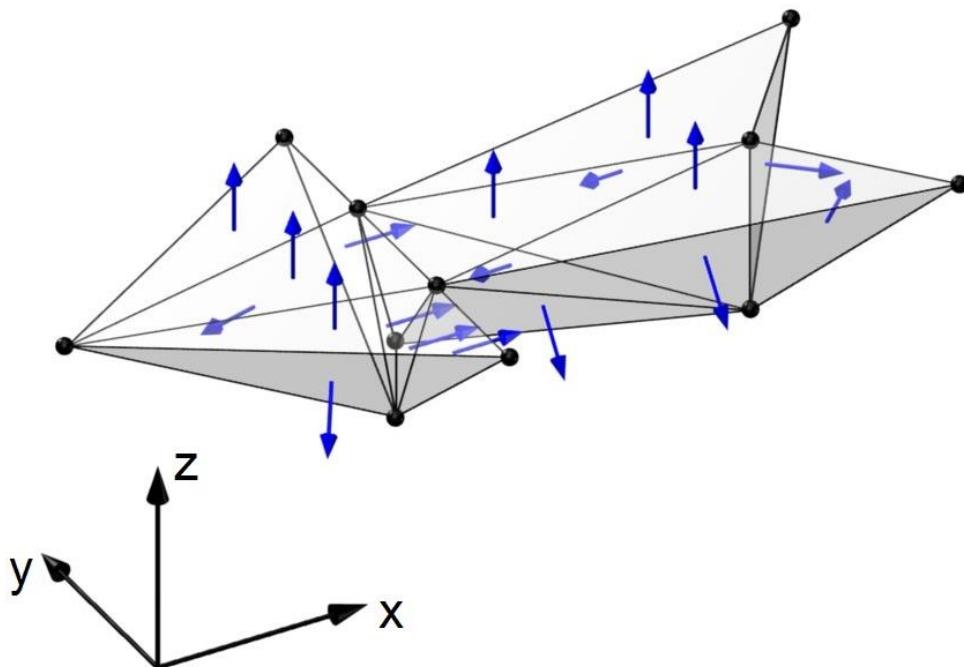


Abbildung 2.8: Stl-Format mit Normalenvektoren

Für jedes Dreieck wird zusätzlich der Normalenvektor gespeichert. Dieser steht immer senkrecht auf der entsprechenden Dreiecksfläche und hat Einheitslänge eins. Dieser Vektor ist wichtig, um zwischen Innen- und Aussenseite des jeweiligen Dreiecks unterscheiden zu können. In Abbildung 2.8 sind die Normalenvektoren durch blaue Pfeile veranschaulicht.

2.7 Slicing

Bei der 2.5D Laserbearbeitung wird das Material Schicht für Schicht von oben nach unten abgetragen. Damit die Maschinensteuerung weiß wie die Schichten aussehen, die abgetragen werden, müssen diese anhand der Geometrie des abzutragenden Volumens berechnet werden. Der Prozess zum Berechnen dieser Schichten nennt sich Slicing.

Beim Slicing wird das dreidimensionale Objekt auf bestimmten Höhen geschnitten. Auf jeder Höhe, auf der ein Schnitt angesetzt wird, entsteht eine entsprechende Schnittkontur. In Abbildung 2.9 ist veranschaulicht, wie auf drei unterschiedlichen Höhen Schnitte durch eine Pfeilgeometrie angesetzt werden. Die daraus entstehenden zweidimensionalen Schnittkonturen in blau, hellblau und grün sind ebenfalls ersichtlich.

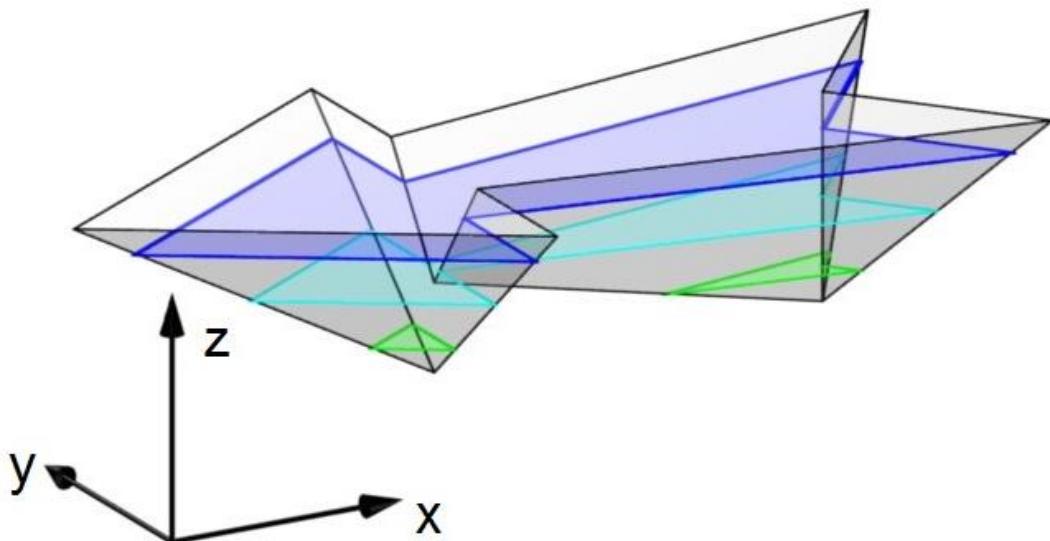


Abbildung 2.9: Beim Slicing entstehen zweidimensionale Schnittkonturen

2.8 Schraffuren

Damit bei der Laserbearbeitung ein Materialabtrag stattfindet, muss der Laser in engen Bahnen innerhalb jeder einzelnen abzutragenden Schicht entlangfahren. Diese eng aneinander liegenden, parallelen Bahnlinien nennt man Schraffuren oder Hatches. Für jede Schnittkontur, wird eine Schraffur berechnet. Dieser Prozess ist in Abbildung 2.10 veranschaulicht.

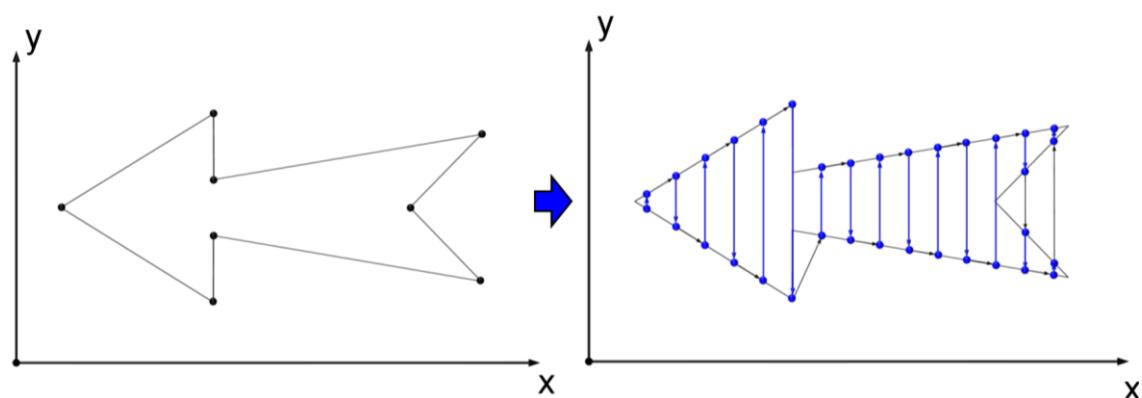


Abbildung 2.10: Die Schraffuren werden aus den geschlossenen Konturen berechnet

2.9 Skywritelinen

Mit der Anwendung von Skywritelinen können bei hohen Scangeschwindigkeiten Abtragsfehler bei der Laserbearbeitung vermieden werden. Die Idee der Skywritelinen ist in einem Artikel [9] von Beat Jaeggi und Beat Neuenschwander beschrieben und wird im Folgenden nochmals zusammengefasst.

2.9.1 Problemstellung

Die Galvomotoren und Spiegel im Scankopf brauchen eine gewisse Zeit, bis sie die gewünschte Geschwindigkeit erreicht haben. Es existiert eine Beschleunigungsphase bis die Scangeschwindigkeit erreicht ist und eine Abbremsphase bis die Scangeschwindigkeit wieder null ist. In Abbildung 2.11 links sind diese Phasen als Rampen dargestellt. Der Laser sendet die Lichtimpulse, die in Abbildung 2.11 links als blaue Kreise visualisiert sind in regelmässigen Abständen auf das Werkstück aus. Da der Laser schon vor der Beschleunigungsphase eingeschaltet wird, landen beim Start und beim Ende einer Bearbeitungslinie mehr Lichtimpulse auf dem Werkstück. An diesen Stellen wird mehr Material abgetragen. Der resultierende Materialabtrag im Profil ist in Abbildung 2.11 rechts veranschaulicht.

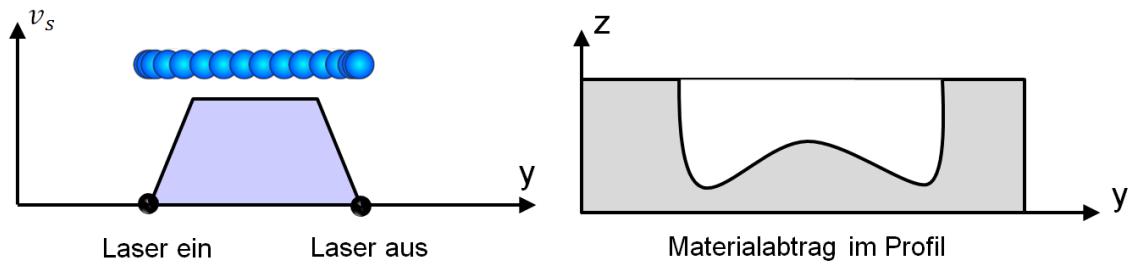


Abbildung 2.11: Laserbearbeitung ohne Skywrite

Werden die Laserbahnen wie in der Abbildung 2.12 ohne Skywrite abgefahren, treffen entlang der Kontur mehr Laserimpulse auf. An diesen Stellen wird mehr Material abgetragen.

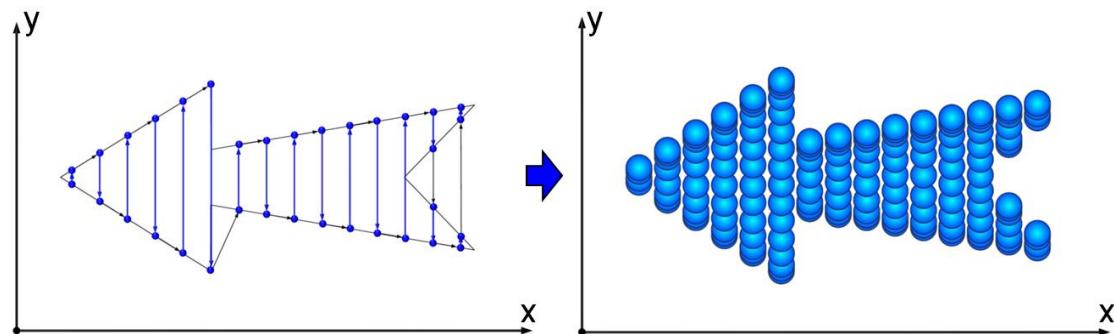


Abbildung 2.12: Ohne Skywrite resultiert ein erhöhter Materialabtrag entlang der Kontur

2.9.2 Lösungsansatz

Die Lösung für dieses Problem sind sogenannte Skywritelinen. Diese werden vor und hinter jeder einzelnen Laserbearbeitungslinie angehängt. In Abbildung 2.13 links sind diese hellblau dargestellt. Auf diesen Linien hat der Scankopf Zeit, um die Galvomotoren mit Spiegel zu beschleunigen oder abzubremsen.

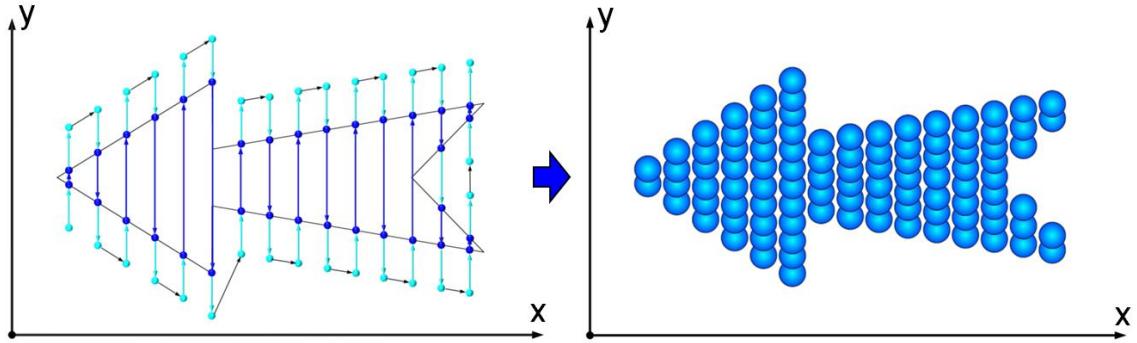


Abbildung 2.13: Mit Skywrite ist der Materialabtrag gleichmässig

Wenn diese die benötigte Scangeschwindigkeit erreicht haben und sich der Fokuspunkt über der Kontur befindet, wird der Laser eingeschaltet (siehe Abbildung 2.14 links). Der Fokus des Lasers fährt dann mit konstanter Geschwindigkeit und konstantem Abtrag über die Kontur. Der Materialabtrag im Profil, wie er in Abbildung 2.14 rechts dargestellt ist, wird gleichmässiger.

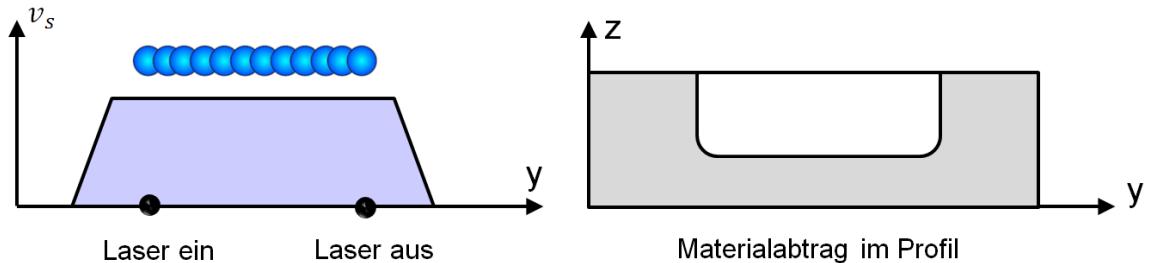


Abbildung 2.14: Laserbearbeitung mit Skywrite

2.9.3 Abschätzung der Skywritelängen

Michael Dietlicher hat in seiner Masterarbeit [10] mit dem Scankopf hurrySCAN II Messungen zur Berechnung der Beschleunigung der optischen Achsen gemacht. In seiner Arbeit wurde folgende gemittelte Beschleunigung berechnet:

$$a_{opt} = 2'4632 \text{ m/s}^2 \quad (2.1)$$

Mit Formel (2.2) kann mit der Beschleunigung a_{opt} und der Scangeschwindigkeit v_s die Beschleunigungstrecke s berechnet werden, die nötig wird um aus dem Stillstand die Scangeschwindigkeit v_s zu erreichen.

$$s = \frac{v_s^2}{2 \cdot a_{opt}} \quad (2.2)$$

Sicherheitshalber sollte die Skywritelänge grösser als die Beschleunigungstrecke s gewählt werden. Multipliziert man Formel (2.2) mit einem Sicherheitsfaktor 2, ergibt sich Formel (2.3) zur Abschätzung der Skywritelänge l .

$$l = 2 \cdot s = \frac{v_s^2}{a_{opt}} \quad (2.3)$$

2.10 Die Rillenbildung

Ein weiterer Fehler, der bei der Laserbearbeitung entstehen kann, ist die Rillenbildung. Durch ständig wiederholte, vertikale Laserbahnen von Schicht zu Schicht entstehen beim Abtrag Rillen parallel zur Scanrichtung. Der Materialabtrag im Profil, wie er in Abbildung 2.15 dargestellt ist, veranschaulicht dieses Problem.

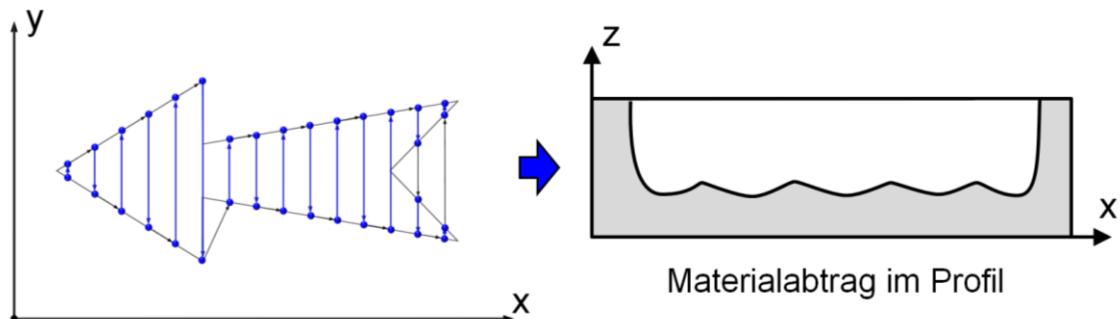


Abbildung 2.15: Rillenbildung

Durch einen sich von Schicht zu Schicht ändernden Schraffurwinkel kann die Rillenbildung reduziert werden. Die Wahl dieses Winkels hat einen Einfluss auf das Ergebnis.

Wenn man einen ändernden Schraffurwinkel von 90 Grad wählt, ist die Richtung der Bahnlinien jeder zweiten Schraffur identisch. Dadurch wird auch das Material wiederholt in diese Richtungen abgetragen und es entstehen rechtwinklig zu einander versetzte Rillen im Werkstück.

Allgemein sind ändernde Schraffurwinkel, die einem Teiler von 360 Grad entsprechen ungünstig. Bei diesen Winkeln kann es vorkommen, dass mehrere Schraffuren dieselbe Richtung haben.

Optimal ist es, wenn man den ändernden Schraffurwinkel als Primzahl wählt. Dadurch ist die Wahrscheinlichkeit kleiner, dass sich eine spätere Schraffurrichtung mit einer vorangehenden überlagert.

In Abbildung 2.16 wurde ein variabler Schraffurwinkel von 13 Grad gewählt. Dadurch wird vermieden, dass die einzelnen Bahnlinien der verschiedenen Schraffuren in dieselbe Richtung zeigen und sich bei der Laserbearbeitung Rillen bilden.

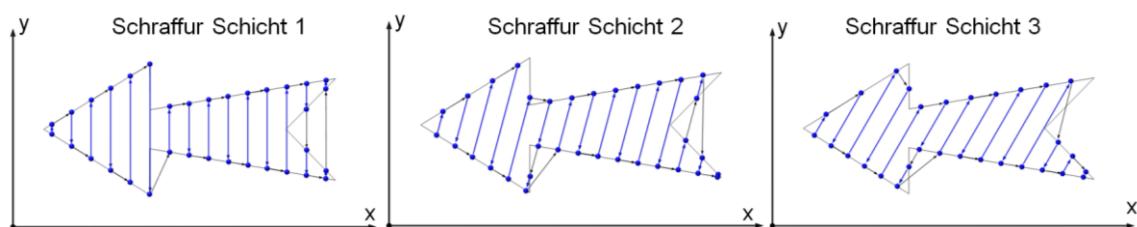


Abbildung 2.16: Variabler Schraffurwinkel

2.11 Strahlkompenstation

2.11.1 Problemstellung

Da der Strahldurchmesser des Lasers im Fokus grösser als null ist, wird die ursprünglich berechnete Schnittkontur bei der Bearbeitung mit dem Laser überschritten. Die ursprüngliche Schnittkontur ist in Abbildung 2.17 rechts mit schwarzen Linien dargestellt. Die durch die

Bearbeitung resultierende Aussenkontur wird dadurch zu gross. Diese ist in Abbildung 2.17 rechts durch die blauen Linien dargestellt. Für präzise Mikrobearbeitung ist das ungenügend.

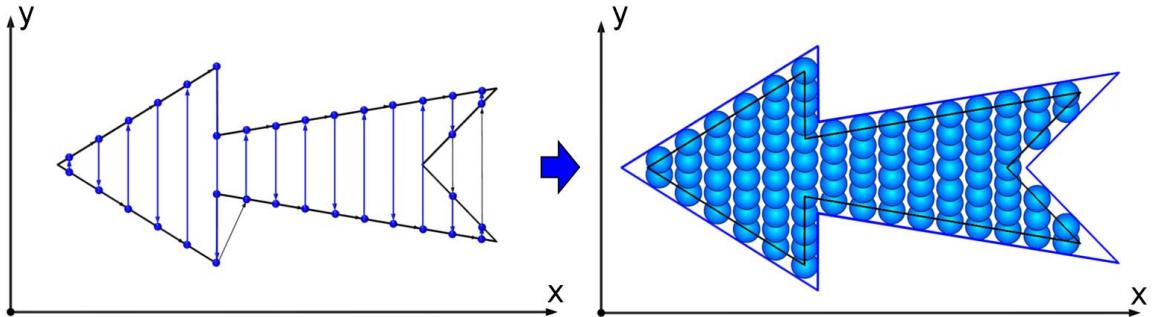


Abbildung 2.17: Laserbearbeitung ohne Strahlkompensation

2.11.2 Lösungsansatz

Mit der Strahlkompensation kann dieses Problem gelöst werden. Dabei wird jede einzelne Kante der berechneten Schnittkontur um einen konstanten Abstand nach innen verschoben. In Abbildung 2.18 wird dieser Prozess veranschaulicht.

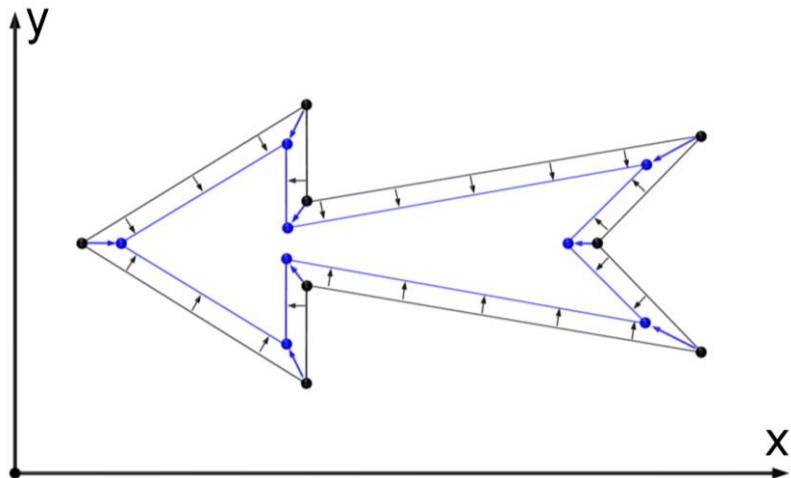


Abbildung 2.18: Berechnung einer neuen, kleineren Kontur

Dadurch entsteht eine neue kleinere Kontur mit neuen Eckpunkten. Mit dieser kleineren Kontur wird dann die Schraffur berechnet. Wenn diese Schraffurlinien vom Laser abgefahren werden, liegt die durch die Bearbeitung erhaltene Aussenkontur genau auf der ursprünglich berechneten Schnittkontur (siehe Abbildung 2.19).

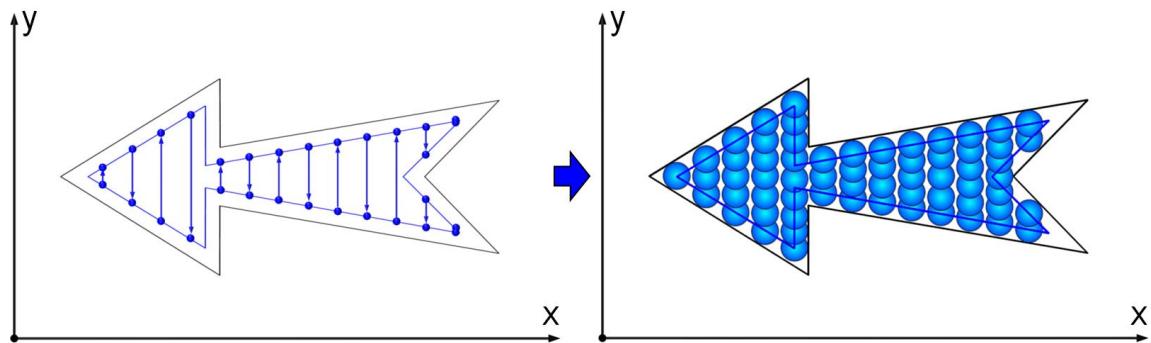


Abbildung 2.19: Laserbearbeitung mit Strahlkompensation

2.11.3 Der Konturabstand bei der Strahlkompensation

In der Bachelorarbeit von Christian Dörig [11] wurden mit Ultrakurz-Puls-Lasern Nuten aus Hartmetall- und CVD-Fräsern abgetragen. In seiner Arbeit wurde die gewünschte Breite der Tasche erst erreicht nachdem eine Strahlkompensation mit einem Konturabstand von 15 Mikrometern angewendet wurde. Der Konturabstand wurde so gewählt, weil er damit genau der Hälfte des Strahldurchmessers von 31 Mikrometern entsprach.

2.11.4 Das Problem der Strahlkompensation

Es ist nicht trivial eine Kontur um einen bestimmten Abstand zu verkleinern, um eine neue kleinere Kontur zu erhalten, wie es in Abbildung 2.18 dargestellt, ist. Es können Spezialfälle auftreten, die Schwierigkeiten bereiten.

Abbildung 2.20 gibt einen Vorgeschmack auf einige Spezialfälle, die bei der Strahlkompensation auftreten können. Enge Konturbereiche können komplett verschwinden. Gewisse Konturen teilen sich in neue Teilkonturen. Ebenso können mehrere Teilkonturen zu einer einzigen Kontur verschmelzen.

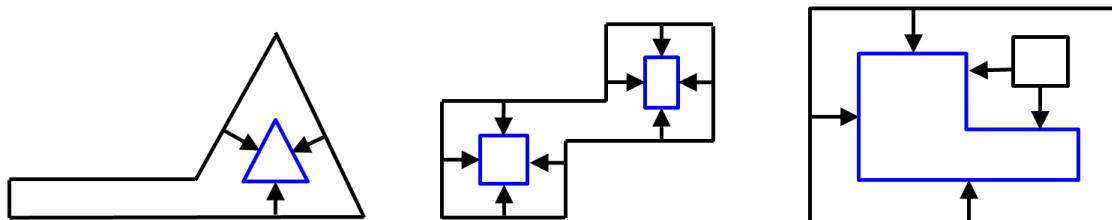


Abbildung 2.20: Bei der Strahlkompensation können unerwartete Konturen entstehen

Zur sauberen Lösung der Strahlkompensation muss man ein Straight Skeleton konstruieren. Die Konstruktion eines Straight Skeleton ist kompliziert. Eine gute Anleitung steht im Internet zur Verfügung [12].

2.11.5 Die Richtung in welche die Kanten verschoben werden

Welche Bereiche innerhalb der Kontur liegen, kann man nicht anhand der Koordinaten und der einzelnen Eckpunkte der geschlossenen Kontur beurteilen. Damit man mit Sicherheit weiß, welcher Bereich innerhalb der Kontur liegt, braucht es zusätzlich die Normalenvektoren der Stl-Datei. Die Normalenvektoren der Stl-Datei sind in Abbildung 2.21 links blau dargestellt. Für jede

Kante der geschlossenen Kontur wird der entsprechende Normalenvektor aus der Stl-Datei gesucht und in die Ebene projiziert, in der die geschlossene Kontur liegt. Die projizierten Normalenvektoren sind in Abbildung 2.21 rechts, blau dargestellt. Entgegen diesen werden die einzelnen Kanten verschoben.

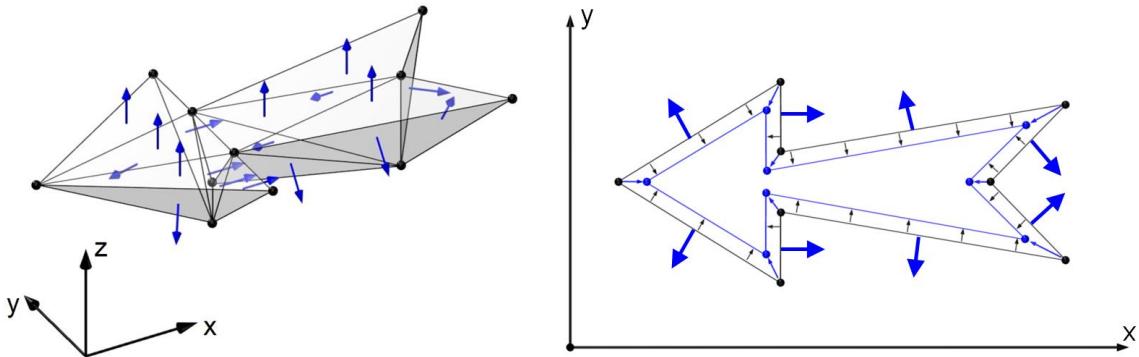


Abbildung 2.21: Verschiebung der Kanten entgegen der Normalenvektoren

2.12 Straight Skeleton

Straight Skeleton heisst auf Deutsch so viel wie Geradengerüst. In Abbildung 2.22 ist ein Straight Skeleton durch die blauen Linien visualisiert. Ein Straight Skeleton kann man sich als Dachkonstruktion vorstellen. Die in Abbildung 2.22 hellblau dargestellte Umrandungskontur repräsentiert dabei die Wände, auf denen das Dach konstruiert wird. Durch schneiden dieser Dachkonstruktion auf bestimmten Höhen entstehen Schnittkonturen. Diese haben immer konstanten Abstand zur Umrandungskontur. Diese Eigenschaft macht das Straight Skeleton so bedeutend. Diese Eigenschaft nimmt man sich bei der Berechnung der Strahlkompensation zu Hilfe.

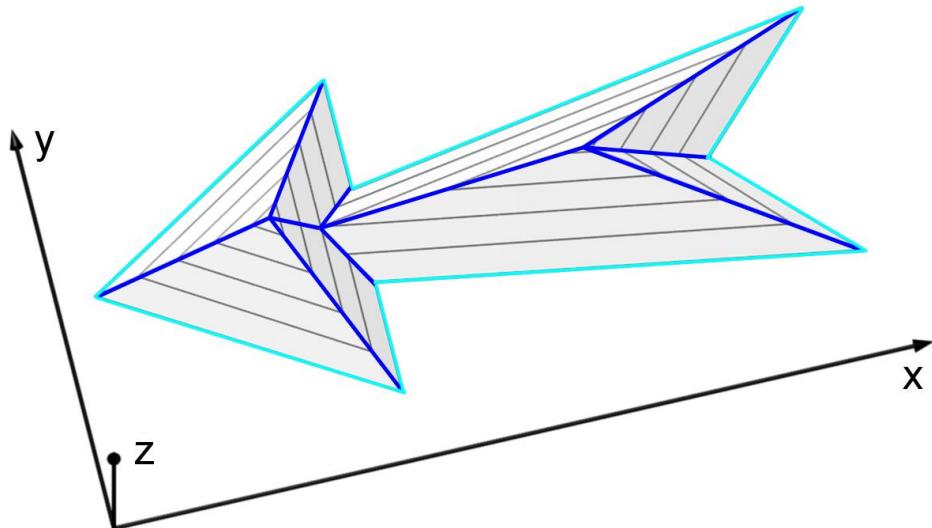


Abbildung 2.22: Veranschaulichung eines Straight Skeleton

2.13 Konturumrandung

Bei der Konturumrandung wird bei der Bearbeitung mit dem Laser die geschlossene Kontur Eckpunkt für Eckpunkt abgefahren, wie es in Abbildung 2.23 links dargestellt ist. Um auch hier einen erhöhten Materialabtrag bei den einzelnen Eckpunkten zu vermeiden, können wie bei den Schraffuren Skywritelinien eingefügt werden. Diese sind in Abbildung 2.23 rechts durch die hellblauen Pfeile verdeutlicht.

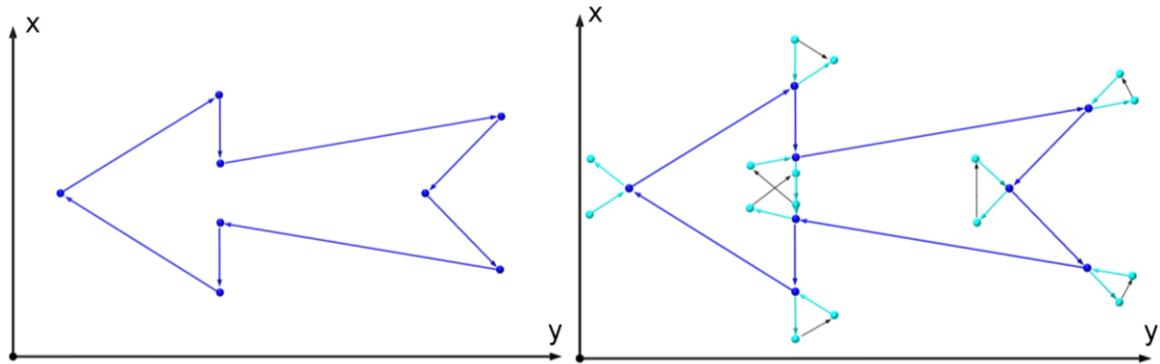


Abbildung 2.23: Konturumrandung links ohne, rechts mit Skywritelinien

2.13.1 Der Breakangle

Der Breakangle β definiert, ab welchem Änderungswinkel α zweier aufeinander folgenden Laserbahnen zusätzlich Skywritelinien eingefügt werden. In Abbildung 2.24 sind zwei Beispiele mit unterschiedlichem Breakangle dargestellt. Im linken Beispiel ist der Änderungswinkel α kleiner als der Breakangle β . Es werden keine Skywritelinien eingefügt. Im rechten Beispiel ist der Änderungswinkel α grösser als der Breakangle β . Deshalb werden hier Skywritelinien berechnet. Diese sind hellblau eingezeichnet.

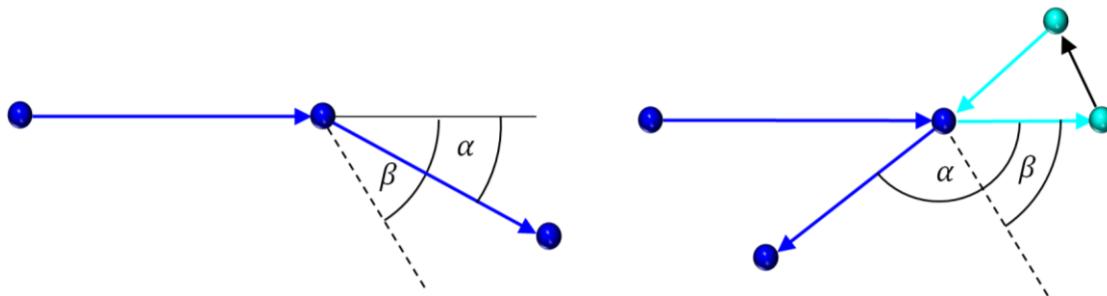


Abbildung 2.24: Skywritelinien werden erst nach Überschreitung des Breakangles eingefügt

2.13.2 Strahlkompensation bei der Konturumrandung

Damit bei der Konturumrandung mit dem Laser die ursprünglich berechnete Schnittkontur nicht überschritten wird, kann diese zuerst mit einer Strahlkompensation verkleinert werden. In Abbildung 2.25 ist die ursprüngliche Schnittkontur schwarz dargestellt. Die Laserbahnen, die bei der Konturumrandung abgefahrene werden, sind blau dargestellt. Damit sich wiederum die Laserbahnen der Umrandung und die Laserbahnen der Schraffur nicht zu stark überschneiden, kann vor der Berechnung der Schraffurlinien eine zweite Strahlkompensation angewendet

werden. Die dadurch erhaltene kleinere Kontur ist in Abbildung 2.25 links hellblau dargestellt. Mit dieser Kontur können die Laserbahnen für die Schraffur berechnet werden.

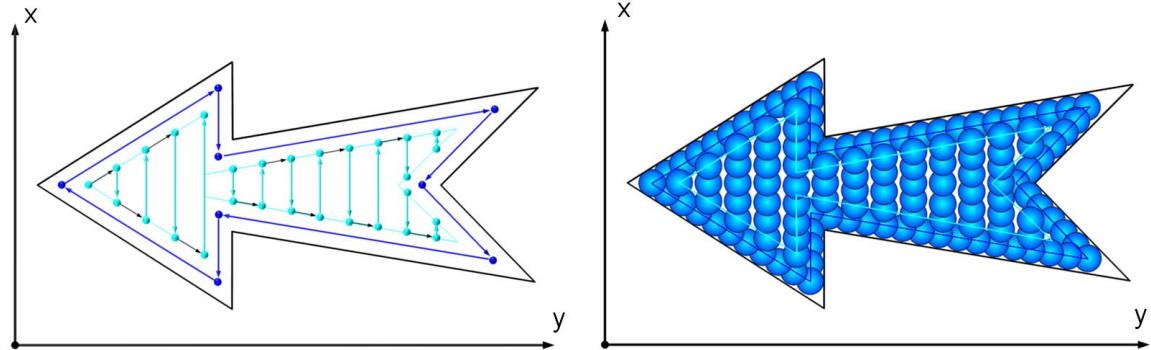


Abbildung 2.25: Strahlkompensation bei der Konturumrandung

In Abbildung 2.25 rechts ist jeder Lichtimpuls des Lasers durch einen blauen Kreis dargestellt. Jeder Lichtimpuls bedeutet Materialabtrag. Mit der zweiten Strahlkompensation kann verhindert werden, dass sich die bearbeiteten Laserbahnen zu stark überlagern und entlang der Kontur zu viel Material abgetragen wird.

3

Versuchsaufbau und Hilfsmittel

3.1 Strahlquelle

Die in dieser Arbeit genutzte Laserquelle ist in Abbildung 3.1 zu sehen. Es handelt sich um einen DUETTO Laser der Firma Time-Bandwidth Products. Die wichtigsten Spezifikationen sind in Tabelle 3.1 aufgelistet.

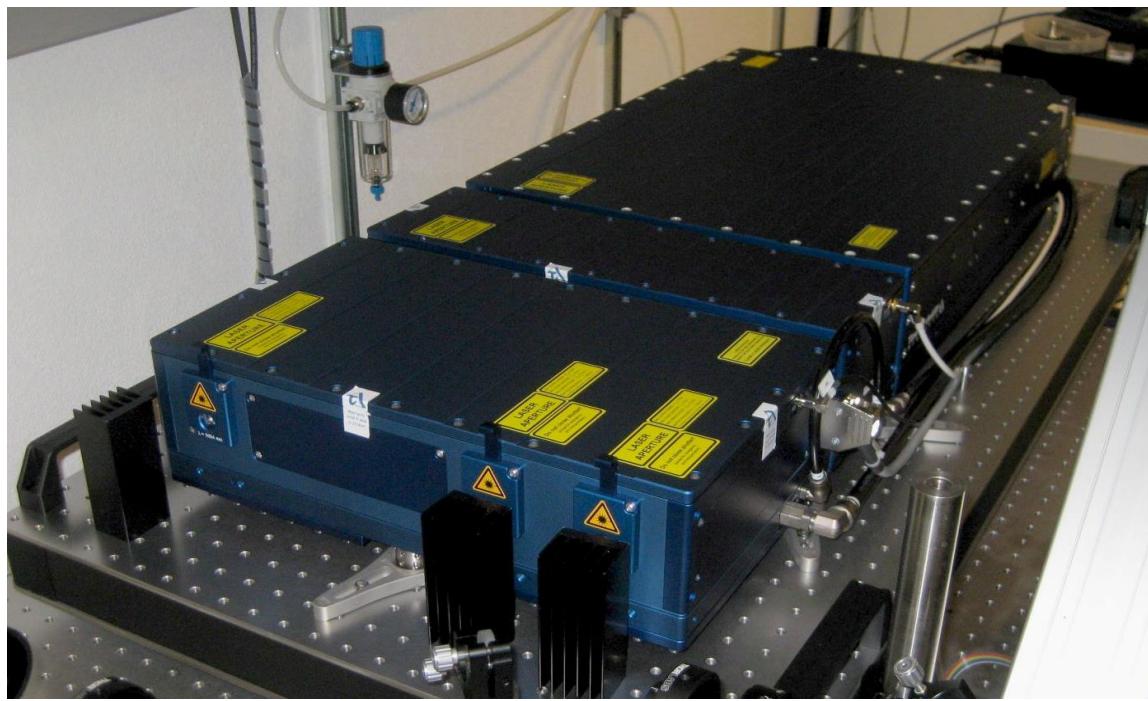


Abbildung 3.1: Der verwendete Ultrakurz-Puls-Laser

Durchschnittliche Laserleistung P_a	10-15 W
Wellenlänge λ	1064 nm
Pulsdauer τ_p	< 12 ps
Pulsfrequenz f_p	50-8200 kHz
Beugungsmasszahl M^2	< 1.3
Pulsenergie E_p	< 200 μ J

Tabelle 3.1: Spezifikationen des DUETTO Lasers [13]

Mithilfe der Beugungsmasszahl M^2 , der Wellenlänge λ , der Brennweite der F-theta Linse f und dem Strahldurchmesser vor dem Scankopf d_s kann man nach Formel (3.4) den Durchmesser des Laserstrahls im Fokus d_f berechnen .

$$d_f = 4 \cdot M^2 \cdot \frac{\lambda}{\pi} \cdot \left(\frac{f}{d_s} \right)^2 \quad (3.4)$$

Nach Formel (3.5) lässt sich die Rayleighlänge z_R berechnen.

$$z_R = 4 \cdot M^2 \cdot \frac{\lambda}{\pi} \cdot \left(\frac{f}{d_s} \right)^2 \quad (3.5)$$

Die doppelte Rayleighlänge bezeichnet man auch als Tiefenschärfe. In diesem Bereich ist mit dem fokussierten Laserstrahl eine Bearbeitung möglich.

3.2 Der Strahlengang

Im Strahlengang wird der gebündelte Laserstrahl aufbereitet und gelangt über mehrere Umlenkspiegel zum Werkstück.

Mit Hilfe von zwei Polarisationsfiltern wird das linear polarisierte Licht der Laserquelle in zirkular polarisiertes Licht umgewandelt. Damit sind die Bearbeitungsergebnisse unabhängig von der Scanrichtung. Diese Filter sind in Abbildung 3.2 links ersichtlich.

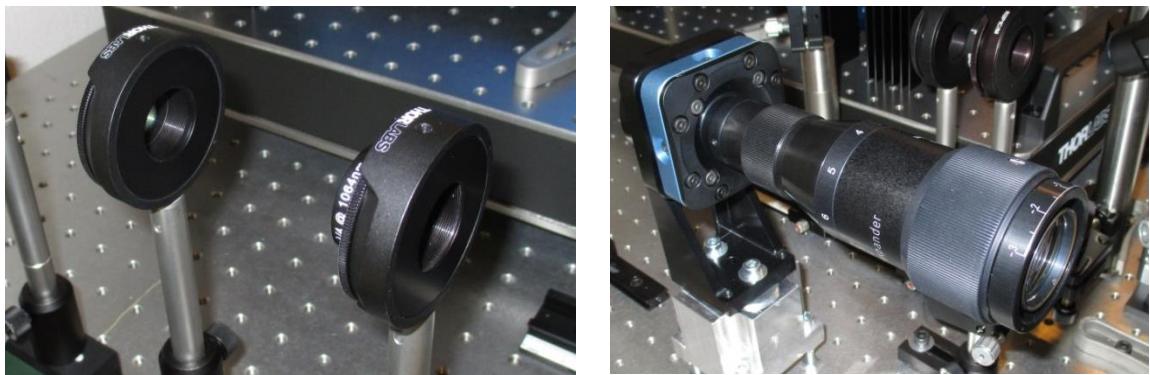


Abbildung 3.2: Optische Instrumente im Strahlengang

Nach Formel (3.4) ist ersichtlich, dass mit einer Vergrösserung des Strahldurchmessers vor dem Scankopf d_s der Strahldurchmesser im Fokus d_f verringert werden kann. Aus diesem Grund wird im Strahlengang der gebündelte Laserstrahl mithilfe eines Teleskops auf einen grösseren Durchmesser auf geweitet. Dieses ist in Abbildung 3.2 rechts zu sehen.

3.3 Die mechanischen und optischen Achsen

In Abbildung 3.3 sind die verschiedenen Achsen abgebildet mit denen die Werkstücke gefertigt wurden. Der Laserstrahl ist zur besseren Darstellung blau eingezeichnet. Dieser gelangt über mehrere Umlenkspiegel in den Scankopf. Verwendet wird der Scankopf hurrySCAN II 14 der Firma SCANLAB. Dieser ist in Abbildung 3.3 durch die Nummer 1 gekennzeichnet.

Das Werkstück wird an der richtigen Stelle auf dem XY-Kreuztisch (Nummer 2 Abbildung 3.3) positioniert.

Der Scankopf ist über eine mechanische Achse (Nummer 3 Abbildung 3.3) am Turm (Nummer 4 Abbildung 3.3) befestigt. Mit dieser Z-Achse kann die Höhe des Scankopfs eingestellt werden. Damit lässt sich der Laserstrahl präzise auf die Oberfläche des Werkstücks fokussieren.

Die mechanische Z-Achse ist eine PRO165 von der Firma Aerotech.

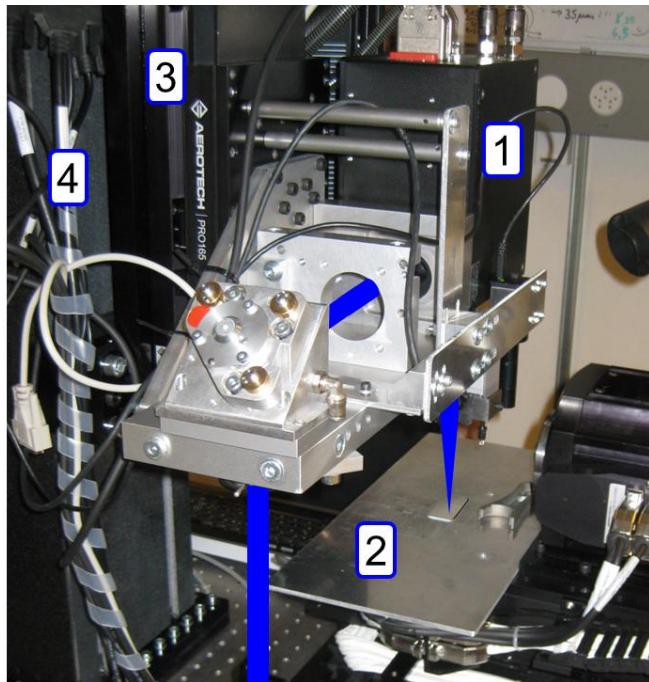


Abbildung 3.3: Der Scankopf, die mechanische Z-Achse und der Werkzeugtisch

3.4 CNC Steuerung

Um den Scankopf anzusteuern, wird der Kontroller Nmark SsaM verwendet. Jede mechanische Achse hat einen weiteren Kontroller. Sie werden über den Computer mit der Software Motion Composer angesteuert. Diese Produkte sind von der Firma Aerotech und bilden zusammen die CNC-Steuerung A3200. Mit diesem Setup lassen sich bis zu 32 Achsen durch mehrere parallel laufende Programme ansteuern [14].

Der Laser und die Achsen können über die Software mit NC-Befehlen nach ISO 6983 [15] angesteuert werden. Trotzdem sind einzelne NC-Befehle wie bei allen CNC-Steuerungen produktbezogen. Die in diesem Kapitel erwähnten NC-Befehle, sind spezifisch für die CNC-Steuerung A3200 von Aerotech. In Kapitel 4.8.3 ist erläutert, wie der NC-Code für diverse Maschinen angepasst werden kann.

3.4.1 Übersicht des verwendeten NC-Codes

Tabelle 3.2 gibt eine Übersicht der wichtigsten NC-Befehle wie sie in dieser Arbeit verwendet wurden. Die optische X-Achse wird mit dem Kürzel U angesprochen, die optische Y-Achse mit dem Kürzel B.

NC-Befehl für CNC-Steuerung A3200	Interpretation
G00 U1 B2	Im Eilgang Position X=1mm, Y=2mm anfahren
G08 G01 U2 B3	Mit definierter Scangeschwindigkeit Position X=2mm, Y=3mm anfahren
GALVO LASEROVERRIDE U ON	Laser anschalten
GALVO LASEROVERRIDE U OFF	Laser ausschalten

Tabelle 3.2: Die wichtigsten NC-Befehle für die CNC-Steuerung A3200 von Aerotech

3.4.2 Übersetzung der einzelnen Bahnlinien in NC-Code

In dieser Arbeit wurden vier Typen von Bahnlinien verwendet. Eilganglinien, Skywrite Startlinien, Skywrite Endlinien und Linien, die mit eingeschaltetem Laser abgefahren werden.

Eilganglinien werden auf der Maschine mit einem G00 Befehl und ausgeschaltetem Laser angesteuert. Skywrite Startlinien werden mit ausgeschaltetem Laser und einem G01 Befehl angesteuert. Ebenso Skywrite Endlinien.

Linien, die mit eingeschaltetem Laser abgefahrene werden, werden ebenfalls mit einem G01 Befehl angesteuert. Vor diesen folgt der entsprechende NC-Befehl zum Einschalten des Lasers. Ist diese Linie zu Ende, wird der Laser mit dem entsprechenden NC-Befehl ausgeschaltet.

3.4.3 NC-Befehle zu Beginn des Codes

Zu Beginn des NC-Codes werden gewisse Befehle einmalig ausgeführt. Diese gelten dann für die gesamte Bearbeitung. Die in dieser Arbeit verwendeten Startbefehle sind in Tabelle 3.3 erläutert.

NC-Befehl für CNC-Steuerung A3200	Interpretation
G90	Absolut Bemassung
VELOCITY ON	Zwischen den einzelnen G01 Befehlen wird die Scangeschwindigkeit beibehalten
AIR_ON	Luftabsaugung einschalten
F1000	G01 Befehle werden mit Scangeschwindigkeit von 1000mm/s ausgeführt

Tabelle 3.3: Vorbereitung des NC-Codes

3.4.4 Ende des NC-Codes

Wie zu Beginn, so braucht es am Ende des NC-Codes einige Befehle, diese sind in Tabelle 3.4 aufgelistet

NC-Befehl für CNC-Steuerung A3200	Interpretation
AIR_OFF	Luftabsaugung ausschalten
END PROGRAM	Am Ende des NC-Codes angelangt

Tabelle 3.4: Ende des NC-Codes

3.4.5 Ausführen von langem NC-Code

Der NC-Code für die Laserbearbeitung kann mehrere Millionen Zeilen Code beinhalten. NC-Code von dieser Länge kann auf der CNC-Steuerung von Aerotech nur mit den Programmbeispiel „BUFFREDRUN“ ausgeführt werden. Mit diesem Befehl wird vermieden, dass der NC-Code in den Arbeitsspeicher des Computers geladen wird und dieser überlastet ist. Stattdessen wird der NC-Code direkt aus der gespeicherten Textdatei, welche den Code beinhaltet, ausgeführt. Im Anhang ist eine Anleitung, die beschreibt wie dieser Befehl angewendet wird.

3.5 Matlab

Matlab ist eine kommerzielle Software des Unternehmens The MathWorks, Inc. Matlab ist primär für numerische Berechnungen mithilfe von Matrizen ausgelegt, woher sich auch der Name ableitet: MATrix LABoratory [16].

Matlab hat den Vorteil, dass es einfach zu benutzen ist. Es gibt eine gute Entwicklungsumgebung, die einem das Finden von Programmierfehlern erleichtert. Zudem ist das grafische Darstellen verschiedener Funktionen einfach realisierbar.

Die geschriebenen Programme werden nur dann schnell ausgeführt, wenn bei der Programmierung auch die Funktionen verwendet werden, die von Matlab zur Verfügung stehen. Zudem muss konsequent mit Vektoren und Matrizen gerechnet werden.

Für Matlab sind nicht so viele Bibliotheken und Funktionen verfügbar wie für C++ oder Java. Deshalb wurden in dieser Arbeit alle Teifunktionen aus denen das CAM Programm aufgebaut ist aus den bereits von Matlab vorgegebenen Funktionen aufgebaut oder neu geschrieben. Einzige Ausnahme ist die Teifunktion zum Importieren der Stl-Dateien in Matlab.

David Gonon, ein Hilfsassistent des Instituts, hat bereits begonnen, ein CAM Programm mit Matlab zu schreiben. In dieser Arbeit konnte auf dessen Code aufgebaut werden. Dies war der entscheidende Grund weshalb mit Matlab gearbeitet wurde.

3.5.1 Stl-Datei in Matlab importieren

Die Funktion mit der die Stl-Dateien in Matlab importiert werden, ist die einzige Teifunktion des CAM Programms, die nicht neu geschrieben wurde. Diese wird vom Unternehmen MathWorks im Internet frei zur Verfügung gestellt [17].

Die Funktion erstellt in Matlab drei Matrizen. In der ersten Matrix werden die X-, Y- und Z-Koordinaten jedes einzelnen Eckpunktes gespeichert. In der zweiten Matrix werden die Informationen gespeichert, aus welchen drei Eckpunkten jeweils ein Dreieck gebildet wird. Die dritte Matrix enthält die X-, Y- und Z-Koordinaten der Normalenvektoren.

Gewisse Bearbeitungsprogramme speichern die Normalenvektoren nicht in der Stl-Datei ab. Die Funktion von MathWorks zum Importieren der Stl-Dateien in Matlab wurde deshalb angepasst. Falls keine oder ungültige Normalenvektoren in der Stl-Datei vorhanden sind werden diese nachträglich berechnet. Diese nachträgliche Berechnung ist möglich anhand der Reihenfolge, in der die Eckpunkte abgespeichert sind. Sind die drei Eckpunkte eines Dreiecks im

Gegenuhrzeigersinn abgespeichert, zeigt der Normalenvektor nach vorne (siehe Abbildung 3.4 links). Sind die drei Eckpunkte im Uhrzeigersinn abgespeichert, zeigt der Normalenvektor nach hinten (siehe Abbildung 3.4 rechts).

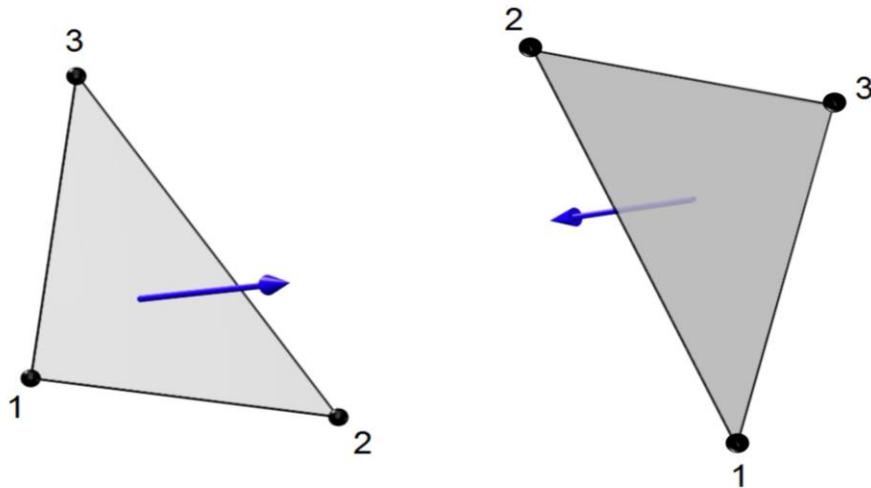


Abbildung 3.4: Nachträgliche Berechnung des Normalenvektors

3.6 Oberflächenrauheit

Die Rauheitsmessungen in dieser Arbeit wurden nach DIN EN ISO 4288 durchgeführt. Diese definiert eine Mindestlänge der Profilmesstrecke sowie die Grenzwellenlänge des Profilfilters. In den Messungen wurden Ra-Werte zwischen 0.1 und 2 μm gemessen. Die Norm schreibt für diese Ra-Werte eine Grenzwellenlänge von 0.8 Millimeter und eine Gesamtstrecke von 4 Millimeter vor.

3.7 Optische Messgeräte

Zur Auswertung der gefertigten Werkstücke wurde in dieser Arbeit zwei Lichtmikroskope des Herstellers Leica benutzt. Das Leica M80 hat einen Vergrößerungsfaktor von 7.5 bis 60. Es dient zur einfachen optischen Überprüfung der Werkstücke. Das Leica DCM 3D bietet die Möglichkeit, durch mehrere konfokale Aufnahmen ein topografisches Oberflächenbild zu erstellen [18].

4

CAM Programmierung

Der grösste Bestandteil dieser Arbeit war das Schreiben des Programmcodes für das CAM. Bei der Programmierung wurde darauf geachtet, dass die einzelnen Schritte, die beim ganzen CAM Programm durchlaufen werden in einzelne Funktionen gegliedert sind. Dadurch ist der Code übersichtlicher. Dank dieses modularen Aufbaus, werden allfällige Anpassungen im Programm vereinfacht.

4.1 Übersicht

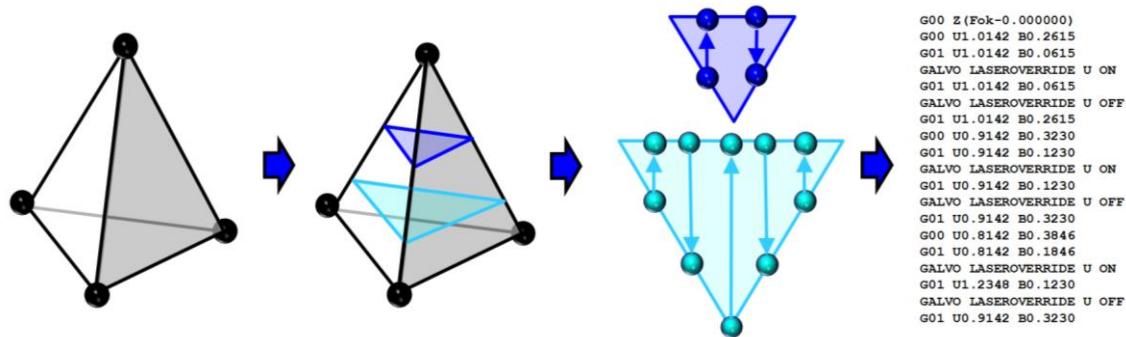


Abbildung 4.1: Vereinfachte Übersicht der CAM Programmierung

In Abbildung 4.1 ist eine vereinfachte Übersicht der CAM Programmierung dargestellt. Als erstes wird das 3D Objekt in Matlab importiert. Dann wird das 3D Objekt auf bestimmten Höhen geschnitten. Auf jeder Höhe, auf der ein Schnitt angesetzt wird, entsteht eine entsprechende Schnittkontur. Im nächsten Schritt werden über diese Schnittkonturen parallele Linien gelegt. Dadurch entsteht für jede Schicht eine eigene, sogenannte Schraffur. Entlang diesen Schraffurlinien muss der Laser später entlangfahren, um das Material Schicht für Schicht abzutragen. Als letztes erfolgt das Postprocessing. Dabei werden die berechneten Schraffuren in NC-Code übersetzt.

4.2 Wahl der Schnithöhen

Es gibt mehrere Möglichkeiten zur Wahl der Schnithöhen, auf denen die dreidimensionale Geometrie geschnitten wird. Eine Variante ist, dass man den obersten Schnitt direkt auf der Oberfläche der Geometrie ansetzt und die folgenden Schnitte jeweils unterhalb um je eine Schichtdicke versetzt (siehe Abbildung 4.2). Dadurch werden die einzelnen Schnittkonturen aber ein bisschen zu gross und bei der Laserbearbeitung wird zu viel Material abgetragen.

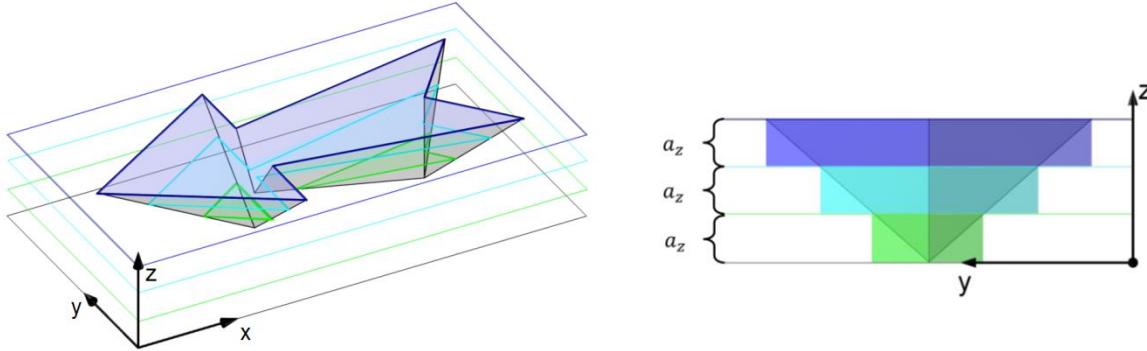


Abbildung 4.2: Der erste Schnitt wird direkt auf der Oberfläche angesetzt.

Eine andere Möglichkeit wäre den ersten Schnitt um eine Schichtdicke versetzt unterhalb der Oberfläche zu wählen und die folgenden Schnitte jeweils um eine Schichtdicke unterhalb darunter versetzt. Dadurch werden die einzelnen Schnittkonturen aber zu klein und bei der Laserbearbeitung wird zu wenig Material abgetragen.

Optimale Schnittkonturen erhält man, wenn der erste Schnitt um eine halbe Schichtdicke versetzt unterhalb der Oberfläche gewählt wird und die folgenden Schnitte jeweils um eine Schichtdicke darunter versetzt. Dadurch haben die Schnittkonturen optimale Grösse und bei der Laserbearbeitung wird der beste Kompromiss für den Materialabtrag erreicht (siehe Abbildung 4.3).

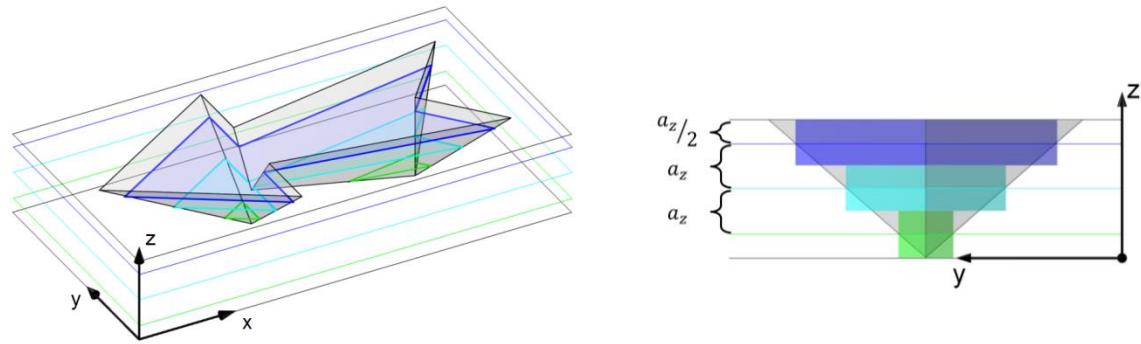


Abbildung 4.3: Der Materialabtrag kommt der Sollgeometrie am nächsten

Erst durch eine geeignete Bestimmung der einzelnen Schnithöhen kann die eigentliche Berechnung der Schnittkonturen erfolgen.

4.3 Berechnung der Schnittkonturen

Die Berechnung der Schnittkonturen gliedert sich in zwei Teilschritte. Im ersten Schritt werden einzelne Schnittpunkte auf den entsprechenden Schnithöhen berechnet. Diese Schnittpunkte

werden im zweiten Schritt zu geschlossenen Konturen verbunden. So ergibt sich für jede Schnitthöhe eine geschlossene Kontur.

4.3.1 Berechnung der Schnittpunkte

Zur Berechnung der Schnittpunkte wird durch jede einzelne Dreiecksfläche iteriert, aus der die Oberfläche der dreidimensionalen Geometrie aufgebaut ist.

Zur Veranschaulichung ist in Abbildung 4.4 eine einzelne Dreiecksfläche dargestellt. Diese wird von drei Schnittebenen auf den entsprechenden Z-Höhen geschnitten. Dadurch entstehen drei Schnittgeraden. Jede dieser Geraden wird durch zwei Schnittpunkte aufgespannt. Die Schnittpunkte entstehen immer dort, wo eine Dreieckskante eine Schnittebene durchstößt. Die Z-Koordinaten dieser Schnittpunkte sind durch die jeweiligen Höhen der Schnittebene bekannt. Da die Koordinaten der Eckpunkte des Dreiecks bekannt sind, können die X- und Y-Koordinaten jedes Schnittpunktes durch lineare Interpolation berechnet werden.

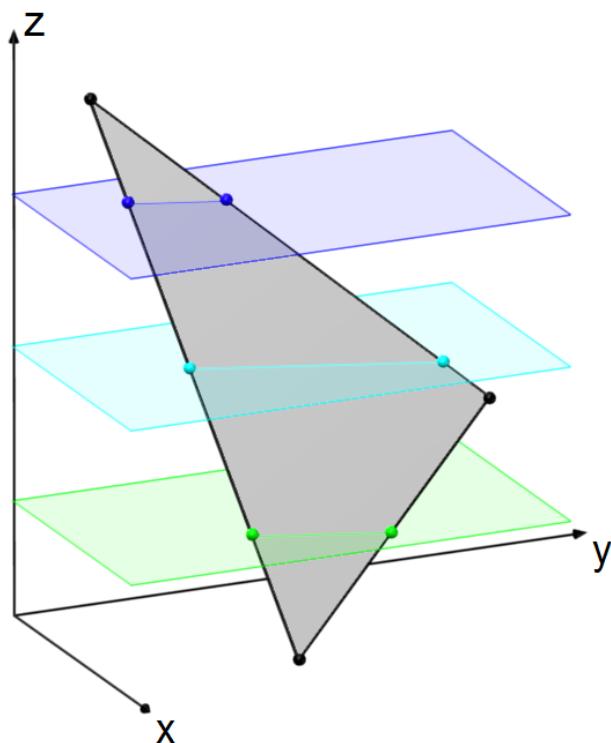


Abbildung 4.4: Darstellung der Schnittgeraden zwischen Dreieck und Schnittebenen

Mit Formel (4.6) kann die X-Koordinate eines Schnittpunkts S berechnet werden, der auf einer Dreieckskante liegt, die von Eckpunkt E1 und Eckpunkt E2 aufgespannt wird. Die Indices geben an um welche Koordinate es sich beim jeweiligen Punkt handelt.

$$S_x = E1_x + \frac{(E2_x - E1_x) \cdot (S_z - E1_z)}{E2_z - E1_z} \quad (4.6)$$

Analog lässt sich für den Schnittpunkt S die Y-Position nach Formel (4.7) berechnen.

$$S_y = E1_y + \frac{(E2_y - E1_y) \cdot (S_z - E1_z)}{E2_z - E1_z} \quad (4.7)$$

Für jedes Dreieck lässt sich ein Set an Schnittpunkten berechnen, welche die einzelnen Schnittgeraden aufspannen. Jeder berechnete Schnittpunkt wird in einer Sammelmatrix zusammen mit der jeweiligen Dreiecksnummer gespeichert.

Die Oberfläche der dreidimensionalen Geometrie ist durch ein geschlossenes Netz von Dreiecken aufgebaut. Da an jede Dreieckskante ein weiteres Dreieck angrenzt, ist letztlich jede Dreieckskante doppelt vertreten. Daraus folgt auch, dass jeder Schnittpunkt genau zwei Mal berechnet wird. Damit der darauf folgende Schritt funktioniert, der die einzelnen Schnittpunkte zu geschlossenen Konturen verbindet, darf jeder Schnittpunkt nur zwei Mal in der Sammelmatrix gespeichert sein. Deshalb müssen ein paar Spezialfälle genauer betrachtet werden. Diese Spezialfälle treten immer dann auf, wenn eine Dreieckskante genau in einer Schnittebene liegt. Abbildung 4.5 veranschaulicht dieses Problem besser.

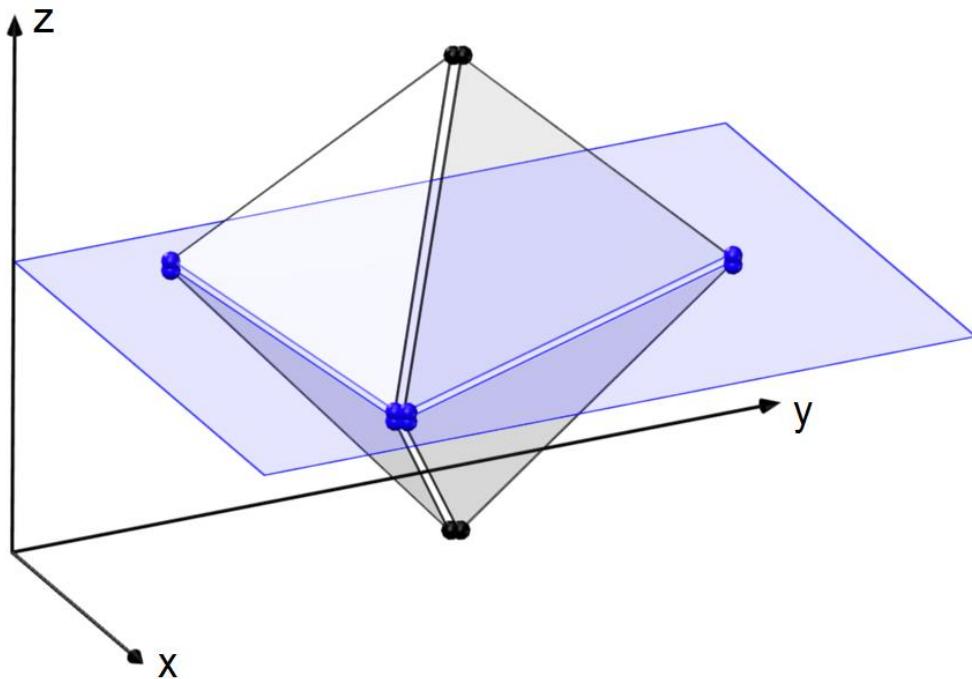


Abbildung 4.5: Mehrere Eckpunkte liegen genau in einer Schnittebene

Falls eine Dreieckskante genau auf Schnithöhe liegt, liegt automatisch eine weitere Dreieckskante auf Schnithöhe. Würde man alle diese Eckpunkte als Schnittpunkte speichern, könnte es vorkommen, dass derselbe Schnittpunkt mehr als zwei Mal in der Sammelmatrix vertreten ist und dadurch beim Verbinden der einzelnen Schnittpunkte zu geschlossenen Konturen Fehler auftreten. Deshalb werden folgende drei Fälle speziell behandelt.

Spezialfall A: Die obere Kante eines Dreiecks liegt genau in einer Schnittebene. In diesem Fall werden die zwei oberen Eckpunkte, die diese Kante aufspannen, nicht gespeichert da diese Punkte bereits durch das darüber liegende Dreieck gespeichert werden. Allfällige Schnittpunkte zwischen anderen Schnittebenen und den anderen zwei Dreieckskanten werden normal durch lineare Interpolation berechnet und in der Sammelmatrix gespeichert.

Spezialfall B: Die untere Kante eines Dreiecks liegt genau in einer Schnittebene. Hier werden die unteren zwei Eckpunkte, die genau in der Schnittebene liegen, gespeichert, da sie nicht durch die darunterliegende Dreiecksfläche gespeichert werden. Falls es Schnittpunkte zwischen den anderen zwei Dreieckskanten und weiteren Schnittebenen gibt, werden diese normal berechnet und in der Sammelmatrix gespeichert.

Spezialfall C: Alle drei Eckpunkte und Kanten eines Dreiecks liegen in einer Schnittebene. In diesem Fall wird keiner der Eckpunkte in der Sammelmatrix gespeichert. Die benötigten Eckpunkte werden durch die angrenzenden Dreiecke bereits richtig in der Sammelmatrix gespeichert.

Falls ein einzelner Eckpunkt eines Dreiecks genau eine Schnittebene berührt, wird dieser Eckpunkt nicht in der Sammelmatrix gespeichert. Ein einzelner Schnittpunkt kann keine Schnittkontur aufspannen und wird verworfen.

Dank dieser Fallanalyse wird jeder benötigte Schnittpunkt genau zwei Mal gespeichert. Im folgenden Schritt treten dadurch keine Fehler auf.

4.3.2 Verbinden der Schnittpunkte zu geschlossenen Konturen

Zunächst werden alle Schnittpunkte in der Sammelmatrix nach Z-Höhe sortiert und in einzelne Matrizen für jede Schnittebene aufgeteilt. So entsteht für jede Schnithöhe eine separate Matrix mit einem Set an Schnittpunkten. Diese separaten Matrizen werden nacheinander ausgewertet.

Auf jeder Schnithöhe erzeugt ein geschnittenes Dreieck eine Schnittgerade. Diese Schnittgerade wird durch zwei Schnittpunkte aufgespannt. Diejenigen Schnittpunkte, die eine Gerade aufspannen, besitzen dieselbe zusätzlich gespeicherte Dreiecksnummer (siehe Abbildung 4.6 links).

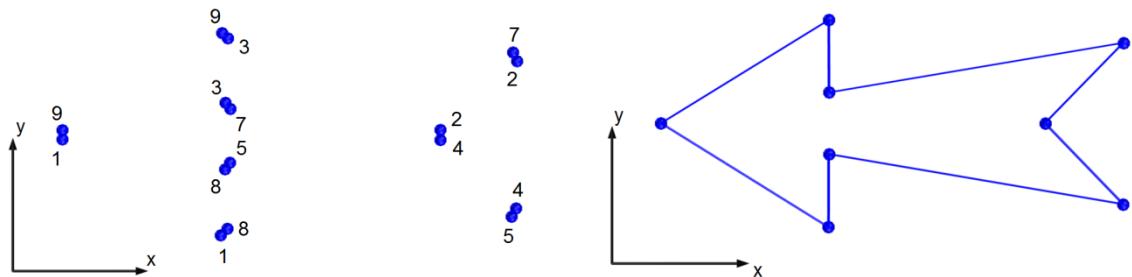


Abbildung 4.6: Die einzelnen Schnittpunkte werden zu Schnittkonturen verbunden

Zudem stoßen zwei benachbarte Geraden immer in einem gemeinsamen Schnittpunkt aneinander. Mit diesen zwei Informationen kann man sich von Punkt zu Punkt durch die Punktwolke hangeln und die einzelnen Punkte in der entsprechenden Matrix in richtiger Reihenfolge sortieren. Nach dem Sortiervorgang kann jeder zweite, doppelt vertretene Punkt gelöscht werden. Werden die sortierten Punkte der Reihe nach abgefahren, entsteht die Kontur, die in Abbildung 4.6 rechts dargestellt ist.

4.4 Berechnung der Schraffuren

Zur Bestimmung der Schraffuren benötigt man die geschlossenen Schnittkonturen. Über jede Kontur wird ein vertikales Linienraster mit konstantem Linienabstand a_b gelegt (siehe Abbildung 4.7). Anschliessend wird durch jede einzelne Kante der geschlossenen Kontur iteriert. Immer dort, wo eine Kante eine vertikale Rasterlinie schneidet, wird durch lineare Interpolation nach Formel (4.9) ein Schnittpunkt S berechnet. Die Indices geben an um welche Koordinate es sich beim jeweiligen Punkt handelt. Die Eckpunkte E1 und E2 spannen die Kante auf, auf der sich Schnittpunkt S befindet. Die X-Koordinate des Schnittpunktes S ist über die X-Koordinate der entsprechenden Rasterlinie gegeben.

$$S_y = E1_y + \frac{(E2_y - E1_y) \cdot (S_x - E1_x)}{E2_x - E1_x} \quad (4.8)$$

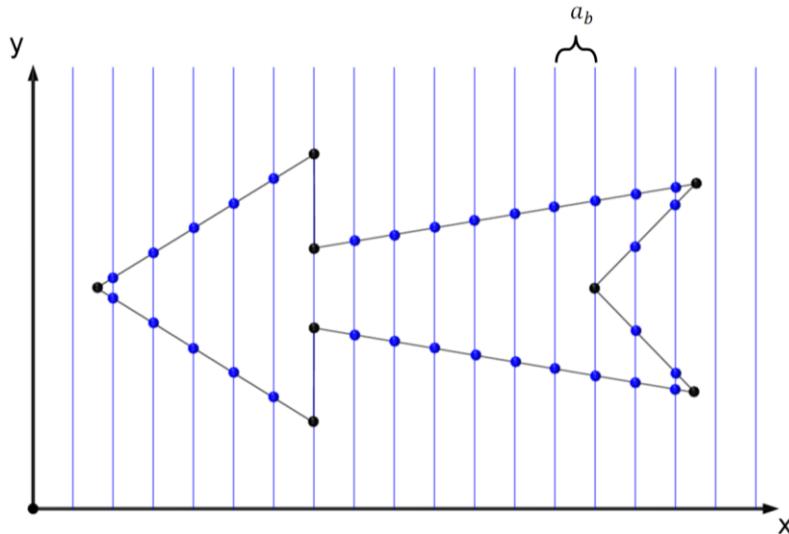


Abbildung 4.7: Berechnung der Schnittpunkte durch das vertikale Linienraster

Die berechneten Schnittpunkte sind in Abbildung 4.7 als blaue Punkte dargestellt und werden in einer Sammelmatrix gespeichert

Die Eckpunkte der geschlossenen Kontur (in Abbildung 4.7 als schwarze Punkte dargestellt), die nicht auf einer Rasterlinie liegen, müssen nicht in der Sammelmatrix gespeichert werden, da sie nicht Teil einer späteren Bahnlinie sind. Jene Eckpunkte der geschlossenen Kontur, die genau auf einer Rasterlinie liegen, müssen speziell geprüft werden. Jede Bahnlinie die mit eingeschaltetem Laser abgefahrene wird, darf nur von zwei Schnittpunkten aufgespannt werden. Einen Startpunkt, um den Laser einzuschalten und einen Endpunkt, um den Laser wieder auszuschalten. Liegen mehr als zwei Schnittpunkte auf einer Laserbearbeitungslinie kann der Laser falsch ein- oder ausgeschaltet werden.

Um diesen Fehler bei der Laserbearbeitung zu vermeiden, werden bei der Schraffurberechnung folgende vier Fälle speziell analysiert.

4.4.1 Spezialfälle bei der Schraffurberechnung

Jeder Spezialfall, der bei der Schraffurberechnung entstehen kann, ist in Abbildung 4.8 aufgeführt.

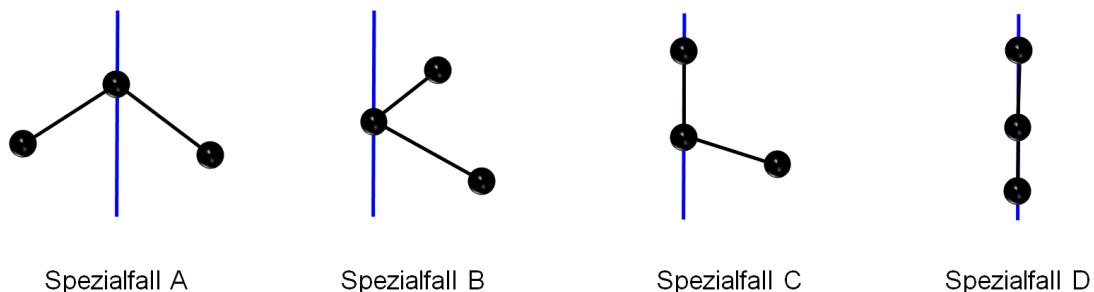


Abbildung 4.8: Spezialfälle, die bei der Schraffurberechnung auftreten können

Spezialfall A: Der Eckpunkt der geschlossenen Kontur, der genau von einer Rasterlinie geschnitten wird, hat einen benachbarten Eckpunkt, der links liegt und einen benachbarten Eckpunkt, der rechts liegt. Er entspricht dem Start- oder Endpunkt einer Bahnlinie und muss deshalb in der Sammelmatrix gespeichert werden.

Spezialfall B: Der Eckpunkt der geschlossenen Kontur, der genau von einer Rasterlinie geschnitten wird, hat beide benachbarten Eckpunkte links liegend oder beide benachbarten Eckpunkte rechts liegend. Aus diesem einen Eckpunkt, der die Rasterlinie nur berührt, wird keine Bahnlinie aufgespannt, da ein zweiter Punkt fehlt. Deshalb wird dieser Eckpunkt nicht in der Sammelmatrix gespeichert.

Spezialfall C: Der Eckpunkt der geschlossenen Kontur, der genau von einer Rasterlinie geschnitten wird, hat einen benachbarten Eckpunkt, der ebenfalls auf dieser Rasterlinie liegt. In diesem Fall muss nur einer der beiden auf der Rasterlinie liegenden Eckpunkte in der Sammelmatrix gespeichert werden. Ansonsten wird die Laserbearbeitungslinie von mehr als zwei Eckpunkten aufgespannt und der Laser wird zu oft ein- oder ausgeschaltet.

Spezialfall D: Der Eckpunkt der geschlossenen Kontur, der genau von einer Rasterlinie geschnitten wird, hat zwei benachbarte Eckpunkte, die ebenfalls auf dieser Rasterlinie liegen. In diesem Fall wird dieser Eckpunkt nicht in der Sammelmatrix gespeichert weil sonst ebenfalls die Laserbearbeitungslinie von mehr als zwei Eckpunkten aufgespannt wird und der Laser falsch ein- oder ausgeschaltet wird.

4.4.2 Verbinden der Schnittpunkte zu Schraffuren

Die berechneten Schnittpunkte können vom Laser in unterschiedlicher Reihenfolge abgefahrene werden. In Abbildung 4.9 sind zwei mögliche Varianten dargestellt. Bei der Variante, die unten links dargestellt ist, werden die einzelnen Schnittpunkte immer von unten nach oben abgefahrene. Dieser Ansatz ist nicht optimal, weil hier zwischen den einzelnen Laserbahnen lange Bahnlinien entstehen, auf denen der Laser ausgeschaltet ist. Daraus resultieren lange Bearbeitungszeiten. Eine bessere Variante ist unten rechts in Abbildung 4.9 dargestellt. Hier werden die einzelnen Schnittpunkte abwechselnd von unten nach oben und dann wieder von oben nach unten angefahren. Die Bahnlinien auf denen der Laser ausgeschaltet ist werden kürzer. Damit wird auch die Bearbeitungszeit verkürzt.

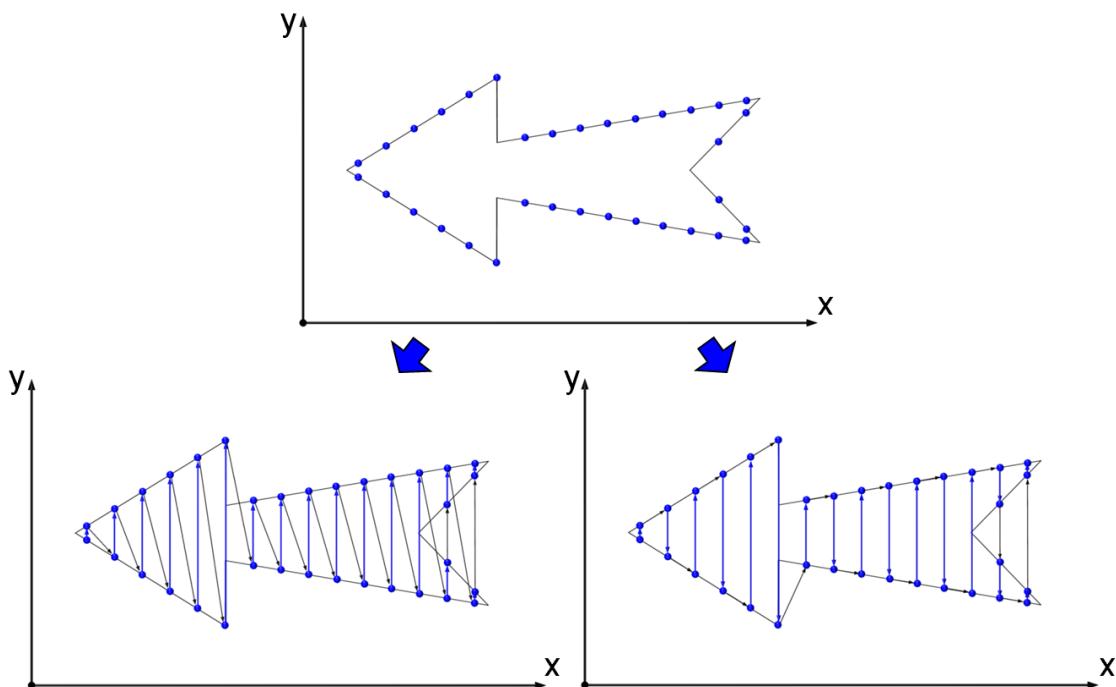


Abbildung 4.9: Die Schnittpunkte können auf verschiedene Weise verbunden werden

4.5 Implementierung Skywrite

Die Implementierung der Skywritelinien ist in Abbildung 4.10 veranschaulicht. Die Schnittpunkte S1 und S2 spannen die Bahnlinie auf, die mit eingeschaltetem Laser abgefahren wird.

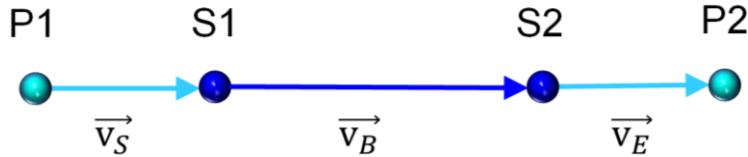


Abbildung 4.10: Berechnung der Skywritelinien

Mithilfe von Formel (4.9) kann der Vektor der Bahnlinie \vec{v}_B berechnet werden.

$$\vec{v}_B = S2 - S1 \quad (4.9)$$

Durch normieren des Vektors \vec{v}_B nach Formel (4.10) erhält man den normierten Bahnvektor \vec{n}_B .

$$\vec{n}_B = \frac{\vec{v}_B}{|\vec{v}_B|} \quad (4.10)$$

Durch Multiplikation des normierten Bahnvektors \vec{n}_B mit der Skywrite Startlänge l_S nach Formel (4.11) erhält man den Skywrite Startvektor \vec{v}_S .

$$\vec{v}_S = \vec{n}_B \cdot l_S \quad (4.11)$$

Dieser wird hinter Schnittpunkt S1 angehängt. Dadurch erhält man mit Formel (4.12) den Punkt P1.

$$P1 = S1 - \vec{v}_S \quad (4.12)$$

Wird der normierte Bahnvektor \vec{n}_B mit der Skywrite Endlänge l_E nach Formel (4.13) multipliziert, erhält man den Skywrite Endvektor \vec{v}_E .

$$\vec{v}_E = \vec{n}_B \cdot l_E \quad (4.13)$$

Dieser wird mit Formel (4.14) vorne an Schnittpunkt S2 angehängt und man erhält den Punkt P2.

$$P2 = S2 + \vec{v}_E \quad (4.14)$$

Alle diese berechneten Punkte werden bei der Laserbearbeitung wie in Abbildung 4.10 dargestellt entsprechend der Vektorpfeile nacheinander abgefahren.

4.6 Implementierung variabler Schraffurwinkel

Die Implementierung des variablen Schraffurwinkels ist in Abbildung 4.11 veranschaulicht. Als erstes werden alle Eckpunkte der geschlossenen Schnittkontur um den entsprechenden Schraffurwinkel δ im Gegenuhrzeigersinn um die Z-Achse gedreht. Dazu nimmt man sich Formel (4.15) zu Hilfe. Diese berechnet aus den X- und Y-Koordinaten des Punktes P1 die neuen X- und Y-Koordinaten des um die Z-Achse gedrehten Punktes P2.

$$\begin{pmatrix} P2_x \\ P2_y \end{pmatrix} = \begin{pmatrix} \cos(\delta) & -\sin(\delta) \\ \sin(\delta) & \cos(\delta) \end{pmatrix} \cdot \begin{pmatrix} P1_x \\ P1_y \end{pmatrix} \quad (4.15)$$

Anschliessend werden die Schnittpunkte zwischen geschlossener Kontur und vertikalem Linienraster berechnet, wie es in Kapitel 4.4 beschrieben ist. Die berechneten Schnittpunkte werden anschliessend um den entsprechenden Schraffurwinkel δ im Uhrzeigersinn um die Z-Achse gedreht. Dazu nimmt man sich die Formel (4.16) zu Hilfe. Diese berechnet aus einem Punkt P_1 und der invertierten Drehmatrix einen neuen um die Z-Achse gedrehten Punkt P_2 .

$$\begin{pmatrix} P_{2x} \\ P_{2y} \end{pmatrix} = \begin{pmatrix} \cos(\delta) & \sin(\delta) \\ -\sin(\delta) & \cos(\delta) \end{pmatrix} \cdot \begin{pmatrix} P_{1x} \\ P_{1y} \end{pmatrix} \quad (4.16)$$

Mit diesem Verfahren erhält man eine Schraffur mit geneigten Schraffurlinien.

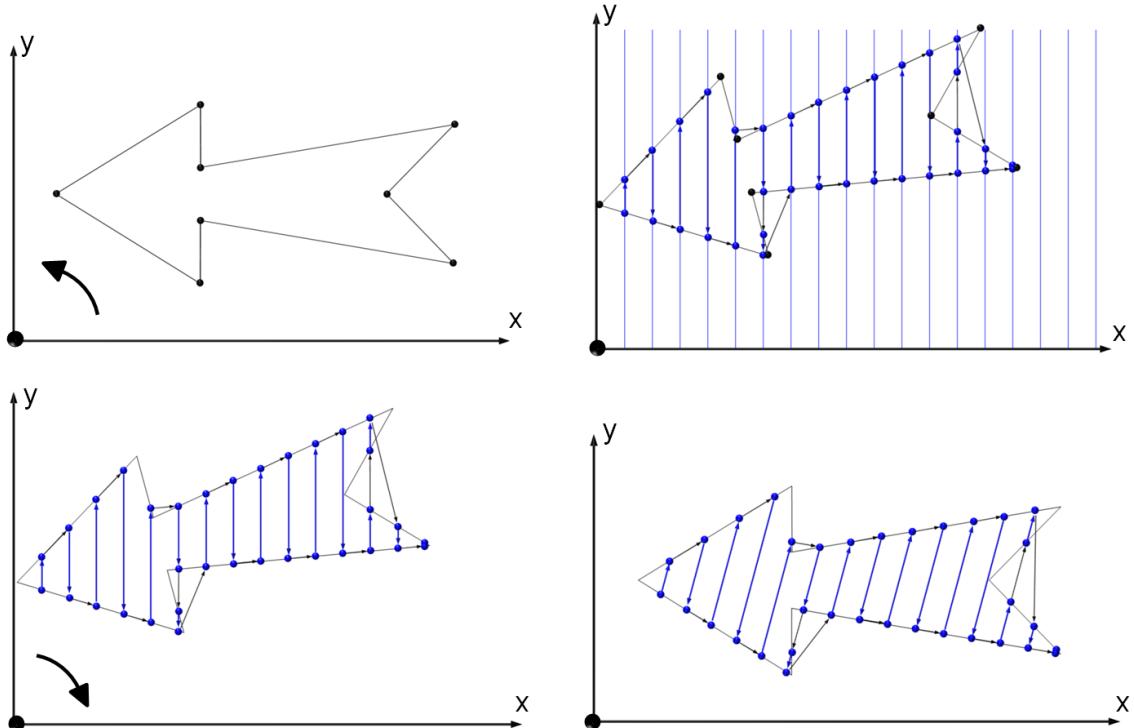


Abbildung 4.11: Berechnung der Schraffurbilder mit geneigtem Bahnlinien

4.7 Strahlkompensation

Von jeder geschlossenen Kontur sind nur die einzelnen Eckpunkte gespeichert. Bei der Strahlkompensation wandern die einzelnen Eckpunkte entlang der Geraden des Straight Skeletons nach innen. Zur Realisierung der Strahlkompensation geht deshalb die Konstruktion des Straight Skeletons einher. Dieses setzt sich aus mehreren Winkelhalbierenden zusammen.

4.7.1 Berechnung der Winkelhalbierenden

Abbildung 4.12 veranschaulicht wie die Winkelhalbierenden berechnet werden.

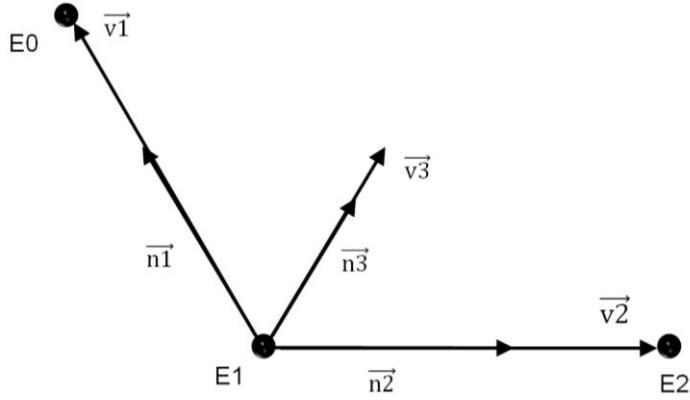


Abbildung 4.12: Berechnung der Winkelhalbierenden

Zur Berechnung der Winkelhalbierenden von Eckpunkt E1, werden die benachbarten Eckpunkte E0 und E2 benötigt. Mit diesen Eckpunkten können nach Formel (4.17) und (4.18) die Vektoren \vec{v}_1 und \vec{v}_2 berechnet werden.

$$\vec{v}_1 = E_0 - E_1 \quad (4.17)$$

$$\vec{v}_2 = E_2 - E_1 \quad (4.18)$$

Diese zwei Vektoren werden mit Formel (4.19) und (4.20) auf Einheitslänge normiert. Daraus resultieren die Vektoren \vec{n}_1 und \vec{n}_2 .

$$\vec{n}_1 = \frac{\vec{v}_1}{|\vec{v}_1|} \quad (4.19)$$

$$\vec{n}_2 = \frac{\vec{v}_2}{|\vec{v}_2|} \quad (4.20)$$

Durch Addition dieser zwei Normalenvektoren nach Formel (4.21), erhält man den Vektor der Winkelhalbierenden \vec{v}_3 .

$$\vec{v}_3 = \vec{n}_1 + \vec{n}_2 \quad (4.21)$$

Wird dieser mit Formel (4.22) normiert, erhält man den Vektor \vec{n}_3 .

$$\vec{n}_3 = \frac{\vec{v}_3}{|\vec{v}_3|} \quad (4.22)$$

4.7.2 Statusberechnung der Eckpunkte

Jeder Eckpunkt der geschlossenen Kontur besitzt einen Status. Alle möglichen Status sind in Abbildung 4.13 dargestellt.

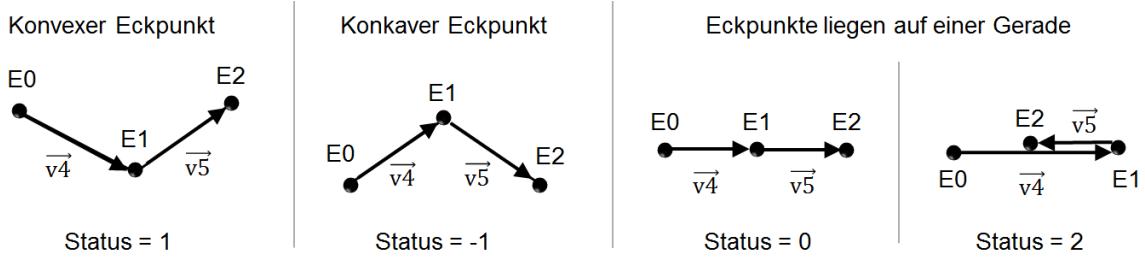


Abbildung 4.13: Berechnung des Status für Eckpunkt E1

Zur Berechnung des Status eines Eckpunktes E1 braucht man die angrenzenden Eckpunkte E0 und E1. Mit Hilfe dieser lassen sich nach Formel (4.23) und (4.24) die Vektoren $\vec{v4}$ und $\vec{v5}$ berechnen.

$$\vec{v4} = E1 - E0 \quad (4.23)$$

$$\vec{v5} = E2 - E1 \quad (4.24)$$

Nach Formel (4.25) kann das Kreuzprodukt k berechnet werden.

$$k = \vec{v4} \times \vec{v5} \quad (4.25)$$

Dieses kann positiv, negativ oder 0 sein. Falls es positiv ist, hat der Eckpunkt E1 Status 1. Es handelt sich um einen konvexen Eckpunkt. Ist das Kreuzprodukt k negativ, hat der Eckpunkt E1 Status -1, es handelt sich um einen konkaven Eckpunkt. Falls das Kreuzprodukt genau 0 ist, muss weiter unterschieden werden. Zeigen beide Vektoren $\vec{v4}$ und $\vec{v5}$ in dieselbe Richtung, hat Eckpunkt E1 Status 0. Zeigen beide Vektoren $\vec{v4}$ und $\vec{v5}$ in entgegengesetzte Richtung hat Eckpunkt E1 Status 2.

4.7.3 Eckpunkte mit Status 0

Eckpunkte die mit ihren benachbarten Eckpunkten auf einer Linie liegen und Status 0 haben, haben bei der Konstruktion des Straight Skeleton keinen Nutzen. Im Gegenteil, sie bedeuten zusätzlichen Ballast im ganzen Algorithmus. Sie können direkt aus der Kontur entfernt werden.

4.7.4 Berechnung des Straight Skeletons bei Konturen mit konvexen Eckpunkten

Bei der Konstruktion des Straight Skeleton wandern die Eckpunkte der geschlossenen Kontur auf den Winkelhalbierenden nach innen. An bestimmten Stellen können die Eckpunkte aufeinander treffen. Dieses Ereignis kann bei konvexen Eckpunkten eintreffen und wird als Edge Event bezeichnet.

Bei einem Edge Event verschmelzen zwei Eckpunkte zu einem neuen. Dieser ist in Abbildung 4.14 rot eingezzeichnet und wird im Folgenden als kritischer Punkt bezeichnet. Die Kante, die zwischen den ursprünglichen Eckpunkten aufgespannt war, verschwindet. Alle anderen Eckpunkte rücken auf und es entsteht eine neue, kleinere geschlossene Kontur. Diese ist in Abbildung 4.14 blau dargestellt. Der kritische Punkt bildet einen Eckpunkt dieser kleineren Kontur. Ausgehend von dieser neuen Kontur beginnt der Prozess von neuem. Die neuen Eckpunkte wandern wieder auf den Winkelhalbierenden nach innen bis sie sich in der Mitte treffen und die Kontur zu einem Punkt zusammengeschrumpft ist. Die Winkelhalbierenden, auf denen die Eckpunkte nach innen wandern, bilden das Straight Skeleton. In Abbildung 4.14 wird dieses durch die schwarzen Pfeile repräsentiert.

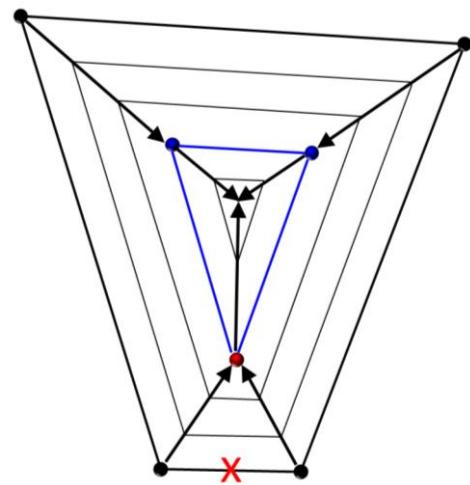


Abbildung 4.14: Bei einem Edge Event verschmelzen zwei Eckpunkte zu einem neuen

4.7.5 Berechnung des kritischen Punktes bei einem konvexen Eckpunkt

Für jeden konvexen Eckpunkt kann ein kritischer Punkt berechnet werden. Von allen kritischen Punkten trifft bei jenem als erstes ein Edge Event ein, der den kürzesten Abstand zur anliegenden Kante hat.

Wie der kritische Punkt des Eckpunktes E1 berechnet wird, ist in Abbildung 4.15 links dargestellt. Als erstes werden die Winkelhalbierenden berechnet. Diese sind in der Abbildung 4.15 hellblau dargestellt. Dann werden zwei Schnittpunkte berechnet. Schnittpunkt S1 befindet sich dort, wo sich die Winkelhalbierende des Eckpunktes E1 und die Winkelhalbierende des benachbarten, linken Eckpunktes E4 schneiden. Der zweite Schnittpunkt S2 liegt dort, wo sich die Winkelhalbierende des Eckpunktes E1 und die Winkelhalbierende des benachbarten, rechten Eckpunktes E2 schneiden. Der kritische Punkt K1 liegt auf jenem Schnittpunkt, der näher beim Eckpunkt E1 ist. In unserem Beispiel in Abbildung 4.15 liegt der kritische Punkt K1 also auf Schnittpunkt S2. Von K1 wird der Abstand A1 zur an Eckpunkt E1 angrenzenden, linken Kante berechnet.

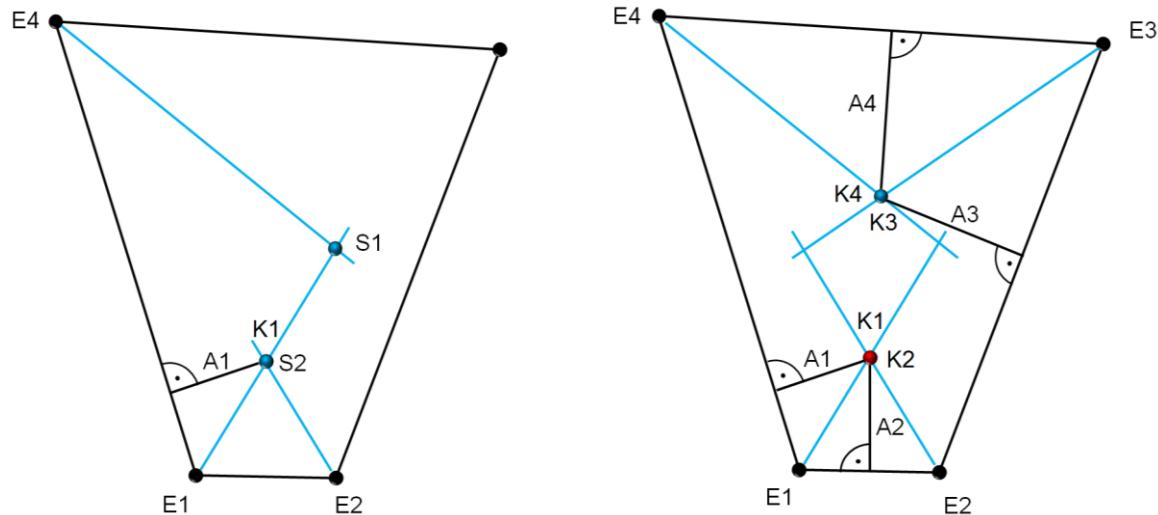


Abbildung 4.15: Bestimmung des kritischen Punktes bei konvexen Eckpunkten

Nach diesem Verfahren wird für jeden Eckpunkt der geschlossenen Kontur ein kritischer Punkt und ein Abstand berechnet. In Abbildung 4.15 rechts sind von jedem Eckpunkt die entsprechenden kritischen Punkte und Abstände eingezeichnet. Abstand A1 und A2 sind die kürzesten, deshalb wird beim kritischen Punkt K1 oder K2 als erstes ein Edge Event eintreffen, wie es in Abbildung 4.14 dargestellt ist.

4.7.6 Berechnung der neuen Kontur bei einem Edge Event

Immer wenn ein Edge Event eintrifft, muss eine neue kleinere geschlossene Kontur gebildet werden. Die neue Kontur hat einen Eckpunkt weniger, weil zwei Eckpunkte der ursprünglichen Kontur zusammentreffen und im kritischen Punkt verschmelzen. Der neu entstandene Eckpunkt hat durch die Koordinaten des kritischen Punktes bereits seine Position in der neuen kleineren Kontur gefunden. Die anderen Eckpunkte der ursprünglichen Kontur müssen um diesen Abstand aufrücken. Wie man die Koordinaten eines neuen aufgerückten Eckpunktes berechnet ist in Abbildung 4.16 dargestellt.

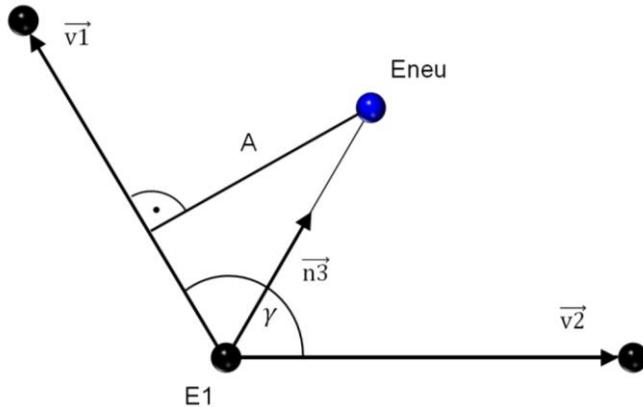


Abbildung 4.16: Berechnung der Position des aufgerückten neuen Eckpunktes

Zur Berechnung der Koordinaten des neuen Eckpunktes benötigt man die Vektoren \vec{v}_1 und \vec{v}_2 . Zusätzlich benötigt man den normierten Vektor der Winkelhalbierenden \vec{n}_3 . Wie man diese drei Vektoren berechnet ist in Kapitel 4.7.1 erläutert. Mit Formel (4.26) lässt sich der Winkel γ zwischen den Vektoren \vec{v}_1 und \vec{v}_2 berechnen.

$$\gamma = \arccos\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1| \cdot |\vec{v}_2|}\right) \quad (4.26)$$

Die Position des aufgerückten Eckpunktes lässt sich mit Formel (4.27) berechnen.

$$E_{neu} = E_1 + \vec{n}_3 \cdot \frac{A}{\sin(\gamma/2)} \quad (4.27)$$

4.7.7 Straight Skeleton Berechnung bei Konturen mit konkaven Eckpunkten

Die Berechnung des Straight Skeleton bei Konturen mit konkaven Eckpunkten, gestaltet sich ähnlich wie jene mit konvexen Eckpunkten. Wieder wandern die Eckpunkte der geschlossenen Kontur entlang den Winkelhalbierenden nach innen. An bestimmten Stellen kann ein nach innen wandernder, konkaver Eckpunkt auf eine ebenfalls nach innen wandernde, gegenüberliegende

Kante treffen. Wenn dieser Fall eintritt, entstehen zwei neue kleinere Teilkonturen. Dieses Ereignis nennt sich Split Event und ist in Abbildung 4.17 veranschaulicht.

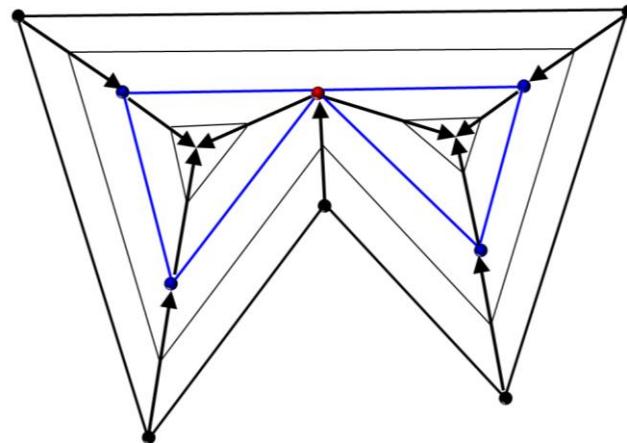


Abbildung 4.17: Bei einem Split Event entstehen zwei neue Teilkonturen

Der Punkt, bei dem der konkave Eckpunkt auf die gegenüberliegende Kante trifft, ist in Abbildung 4.17 rot eingezeichnet. Auch dieser wird im Folgenden als kritischer Punkt bezeichnet. Der kritische Punkt wird so gespalten, dass er in jeder separaten Teilkontur einmal vertreten ist. Die restlichen Eckpunkte der ursprünglichen Kontur wandern ebenfalls um den entsprechenden Abstand nach innen und werden richtig mit dem kritischen Punkt zu neuen, kleineren Teilkonturen verbunden. Diese sind in Abbildung 4.17 blau dargestellt. Für den gespaltenen, kritischen Punkt, der nun in jeder Teilkontur vertreten ist, muss die Winkelhalbierende neu berechnet werden. Anschliessend kann die Straight Skeleton Konstruktion für jede neue Teilkontur fortgesetzt werden. Wieder ergibt sich das gesamte Straight Skeleton aus den Winkelhalbierenden, auf denen die Eckpunkte entlang wandern. In Abbildung 4.17 sind diese durch die schwarzen Pfeile visualisiert.

4.7.8 Berechnung des kritischen Punktes bei einem konkaven Eckpunkt

Für jeden konkaven Eckpunkt kann ein kritischer Punkt berechnet werden. Von allen kritischen Punkten trifft bei jenem als erstes ein Ereignis ein, der den kürzesten Abstand zur gegenüberliegenden Kante hat. Bei Konturen mit konkaven Eckpunkten kann als Ereignis sowohl ein Edge Event als auch ein Split Event eintreffen.

Die Bestimmung des kritischen Punktes für konkave Eckpunkte ist in Abbildung 4.18 veranschaulicht. Als erstes wird die Winkelhalbierende des Eckpunktes E2 berechnet. Danach wird die Winkelhalbierende zwischen der an den konvexen Eckpunkt E2 angrenzende Kante und der gegenüberliegenden Kante berechnet. Diese Winkelhalbierende entspringt im Schnittpunkt S. Die gegenüberliegende Kante ist in Abbildung 4.18 grün eingezeichnet. Dort, wo sich die beiden Winkelhalbierenden schneiden, befindet sich ein kritischer Punkt K.

Der berechnete kritische Punkt K ist nur gültig, falls er innerhalb des Sektors, der von der gegenüberliegenden Kante und den beiden Winkelhalbierenden, welche von den Eckpunkten der gegenüberliegenden Kante ausgehen, liegt. Der eben berechnete kritische Punkt erfüllt diese Bedingung. Der Sektor ist in Abbildung 4.18 rechts hellblau eingefärbt.

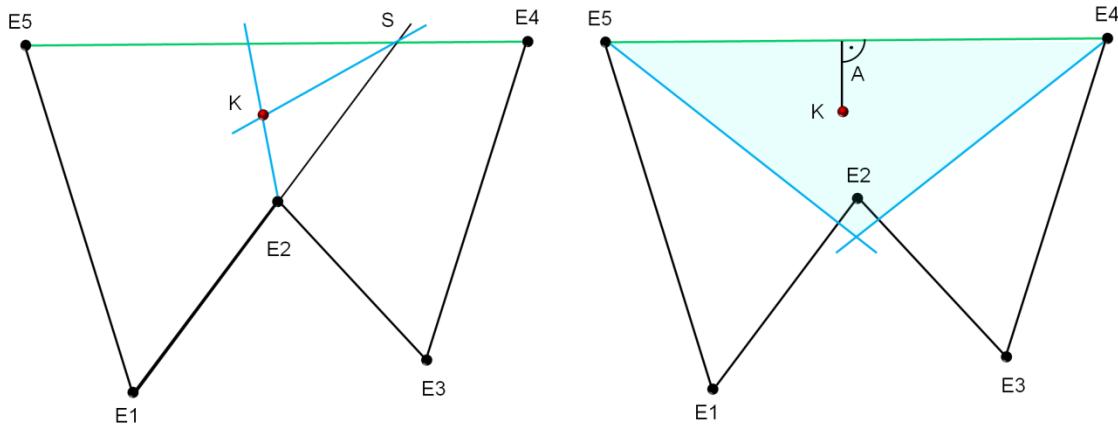


Abbildung 4.18: Berechnung des kritischen Punktes bei einem konkaven Eckpunkt

Jede Kante der geschlossenen Kontur kann die gegenüberliegende Kante sein. Um den richtigen kritischen Punkt zu finden, muss jede Kante der geschlossenen Kontur als gegenüberliegende Kante betrachtet und dafür der kritische Punkt berechnet werden. Von allen gültigen kritischen Punkten ist jener der einzige, der den kleinsten Abstand zum konkaven Eckpunkt E2 hat. Als letztes wird für diesen einzigen kritischen Punkt der Abstand A zur gegenüberliegenden Kante berechnet.

4.7.9 Merge Event

In gewissen Fällen kann es vorkommen, dass bei der Konstruktion des Straight Skeleton zwei Konturen zu einer neuen verschmelzen. Dies trifft immer dann ein, wenn eine innere, kleinere Kontur, die nach aussen wächst, auf eine äussere, grössere Kontur trifft, die von aussen nach innen wächst. Diese Situation nennt sich Merge Event und ist in Abbildung 4.19 illustriert.

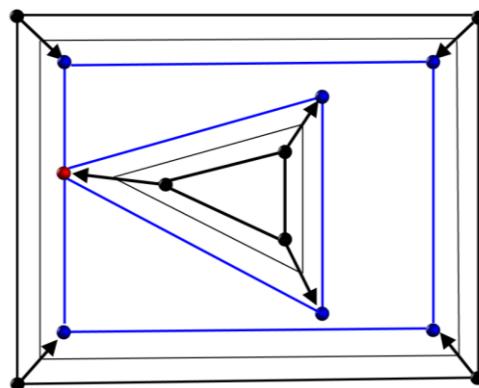


Abbildung 4.19: Bei einem Merge Event verschmilzt die äussere Kontur mit der inneren

Der kritische Punkt, der in Abbildung 4.19 rot eingezzeichnet ist berechnet sich genau gleich wie bei einem Split Event. Auch hier teilt sich der kritische Punkt in zwei neue, die richtig mit den anderen aufgerückten Eckpunkten verbunden werden. Im Gegensatz zum Split Event entstehen hier aber nicht zwei neue Teilkonturen sondern eine einzelne, neue Kontur. Diese ist in Abbildung 4.19 durch die blauen Linien dargestellt.

4.7.10 Korrektur von Konturen mit Eckpunkten vom Status 2

Eckpunkte mit Status 2 treten bei der Straight Skeleton Konstruktion bei Konturen mit rechtwinkligen Ecken und parallelen Kanten auf. Wie der Status der einzelnen Eckpunkte berechnet wird, ist in Kapitel 4.7.2 erläutert.

In Abbildung 4.20 sind drei Konturen dargestellt, bei denen Eckpunkte mit Status 2 auftreten.

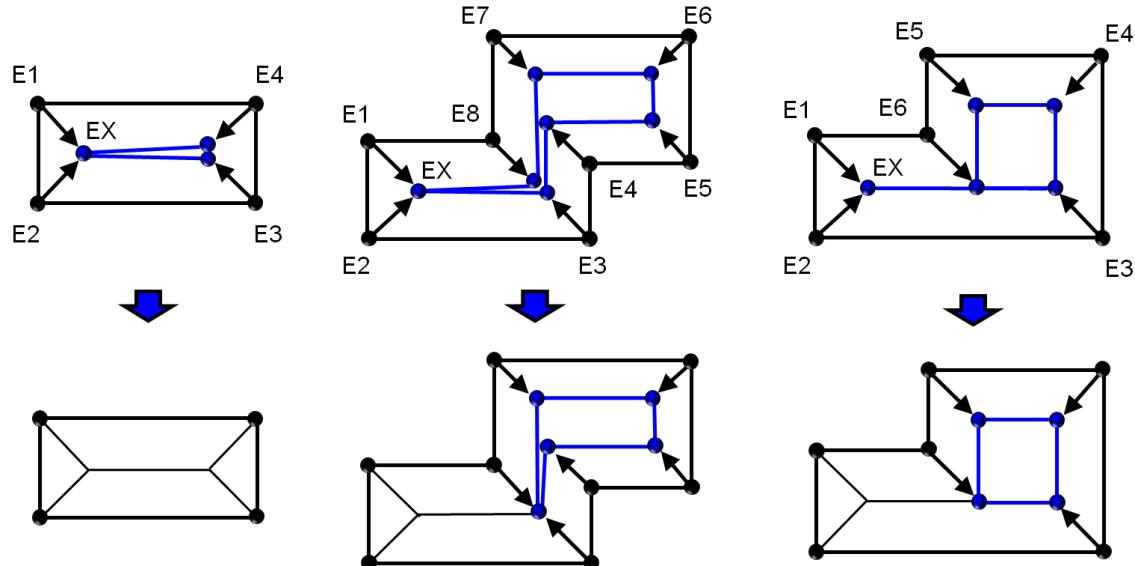


Abbildung 4.20: Eckpunkte mit Status 2 müssen speziell behandelt werden

Immer dort, wo Eckpunkt E1 und Eckpunkt E2 bei einem Edge Event aufeinandertreffen und verschmelzen, liegen die zwei angrenzenden Kanten aufeinander und der neue Eckpunkt erhält Status 2. Dieser Eckpunkt ist jeweils mit EX gekennzeichnet. Damit im Folgenden keine Fehler auftreten, muss dieser Eckpunkt richtig entfernt werden. Im ersten Fall in Abbildung 4.20 links, hat die neue, blau dargestellte Kontur nur noch drei Eckpunkte. Hier wird aus allen drei Eckpunkten eine einzige Straight Skeleton Linie gebildet. Im zweiten Fall in Abbildung 4.20 mittig, sind durch den Edge Event von Eckpunkt E1 und Eckpunkt E2 die benachbarten Eckpunkte E3 und E8 ebenfalls übereinander aufgerückt. In diesem Fall wird der Eckpunkt EX entfernt. Eckpunkt E3 und Eckpunkt E8 verschmelzen zu einem neuen Eckpunkt. Der Status und die Winkelhalbierende dieses neuen Eckpunktes müssen aktualisiert werden. Im dritten Fall, in Abbildung 4.20 rechts, liegen die von EX benachbarten Eckpunkte E3 und E6 nicht aufeinander. Hier wird nur der Eckpunkt EX entfernt. Der Status und die Winkelhalbierende der von EX benachbarten Eckpunkte müssen neu berechnet werden.

4.7.11 Speicherung einer geschlossenen Konturen

Für jede einzelne Kontur wird eine eigene Matrix erstellt. Jede Zeile dieser Matrix enthält die X-, Y- und Z-Koordinaten jedes einzelnen Eckpunktes. Werden diese Koordinaten Zeile für Zeile miteinander verbunden, erhält man die geschlossene Kontur. Damit nicht jedes Mal von neuem der Status und die Winkelhalbierende für jeden Eckpunkt berechnet werden müssen, werden diese Informationen auch in dieser Matrix gespeichert. Spalte vier enthält dabei den Status jedes einzelnen Eckpunktes. Spalte fünf enthält jeweils die X-Komponente und Spalte sechs die Y-Komponente der Winkelhalbierenden.

4.7.12 Implementation des eigenen Straight Skeleton Algorithmus

Die Anleitung zur Berechnung des Straight Skeleton [12] hat bei der Implementation des eigenen Straight Skeleton Algorithmus sehr geholfen. Diese benötigt zur Implementation Pointer und verkettete Listen. In Matlab sind diese Konstrukte schwierig zu realisieren. Eine Implementation nach der Anleitung ist an ihrer Komplexität gescheitert. Deshalb wurde ein eigener Algorithmus geschrieben. Der eigene Algorithmus ist nicht so effizient wie jener, der in der Anleitung vorgeschlagen wurde. Dafür ist er einfacher. Zudem hat der eigene Algorithmus eine zweite Abbruchbedingung. Das heisst, es wird nicht das ganze Straight Skeleton konstruiert bis die geschlossene Kontur zu einem einzigen Punkt zusammengeschrumpft ist, sondern es wird nur soweit konstruiert, bis der benötigte Abstand für die Strahlkompensation erreicht ist. Dadurch erhält man eine Zeitsparnis.

Folgender Pseudocode zeigt, wie der eigene Straight Skeleton Algorithmus für die Strahlkompensation realisiert wurde.

(Initialisierung)

Alle Eckpunkte aller Konturen richtig im Uhrzeigersinn oder Gegenuhrzeigersinn anordnen, damit die linke Seite jeder Kante nach innen zeigt

Status und Winkelhalbierende für alle Eckpunkte berechnen

Zwischenabstand = 0

While = 1

(Teil 1: Vorbereitung)

Durch alle Eckpunkte aller Konturen iterieren

Falls aktueller und darauf folgender Eckpunkt identisch

Aktueller Eckpunkt löschen

Status und Winkelhalbierende des darauf folgenden Eckpunktes aktualisieren

Durch alle Eckpunkte aller Konturen iterieren

Falls Eckpunkt mit Status = 2 gefunden

Falls in der Kontur mit diesem Eckpunkt weniger als 3 Eckpunkte

Kontur löschen

Falls in der Kontur mit diesem Eckpunkt mehr als 3 Eckpunkte

Falls die angrenzenden Eckpunkte nicht identisch

Diesen Eckpunkt löschen

Status und Winkelhalbierenden der angrenzenden Eckpunkte aktualisieren

Falls die angrenzenden Eckpunkte identisch

Diesen Eckpunkt löschen

Einen benachbarten Eckpunkt löschen

Status und Winkelhalbierende des anderen, benachbarten Eckpunkt aktualisieren

Durch alle Eckpunkte aller Konturen iterieren

Falls Eckpunkt mit Status = 0 gefunden

Diesen Eckpunkt löschen

Falls alle Konturen leer (Abbruchbedingung. Die Kontur ist zusammengeschrumpft)

Break

(Teil 2: Kritische Punkte suchen)

Abstandalt = Inf

Durch alle Eckpunkte aller Konturen iterieren

Falls aktueller Eckpunkt Status = 1 hat

 Abstand und kritischer Punkt für konvexen Eckpunkt berechnen

 Abstandneu = Abstand + Zwischenabstand

 Falls Abstandneu < Abstandalt

 Abstandalt = Abstandneu

 Speicherstatus = 1

 Konturaktiv zwischenspeichern. (In dieser Kontur befindet sich der aktuelle Eckpunkt)

Durch alle Eckpunkte aller Konturen iterieren

Falls aktueller Eckpunkt Status = -1 hat

 Abstand und kritischer Punkt für konkave Eckpunkte berechnen

 Abstandneu = Abstand + Zwischenabstand

 Falls Abstandneu < Abstandalt

 Abstandalt = Abstandneu

 Speicherstatus = -1

 Konturaktiv zwischenspeichern. (In dieser Kontur befindet sich der aktuelle Eckpunkt)

 Konturgegenüber zwischenspeichern. (In dieser Kontur befindet sich die gegenüberliegende Kante)

(Teil 3: Neue Kontur bilden)

Falls Konturabstand <= Abstandalt (Abbruchbedingung. Endkontur gefunden)

 Alle Eckpunkte um Konturabstand – Zwischenabstand aufrücken

 Break

Falls Konturabstand > Abstandalt

Falls Speicherstatus = 1 (**Edge Event**)

 Falls weniger als 3 Eckpunkte in Konturaktiv

 Konturaktiv leeren

 Falls nur noch 3 Eckpunkte in Konturaktiv

 (Alle 3 Punkte schrumpfen zu einem Punkt hin und schliessen diese Kontur)

 Konturaktiv leeren

 Falls mehr als 3 Eckpunkte in Konturaktiv

 Alle Eckpunkte um Abstand – Zwischenabstand aufrücken

 Zwischenabstand = Abstandalt

 (Zwei benachbarte Eckpunkte kollidieren im kritischen Punkt)

 Einer dieser Eckpunkte muss aus der Kontur entfernt werden

 Status und Winkelhalbierende des anderen Eckpunktes aktualisieren

Falls Speicherstatus = -1 (Split oder Merge Event)

 Alle Eckpunkte um Abstand – Zwischenabstand aufrücken

 Zwischenabstand = Abstandalt

 Falls Konturaktiv = Konturgegenüber (**Split Event**)

 Der kritische Punkt spaltet sich und ist in jeder neuen Teilkontur einmal vertreten

 Status und Winkelhalbierende von beiden neuen, kritischen Punkten aktualisieren

 Falls Konturaktiv ≠ Konturgegenüber (**Merge Event**)

 Der kritische Punkt spaltet sich und ist in der neuen verschmolzenen Kontur zweimal vertreten

 Status und Winkelhalbierende von beiden neuen kritischen Punkten aktualisieren

Der grosse Bestandteil des Codes bildet eine Whileschlaufe. Diese folgt direkt auf die Initialisierung. Diese Whileschlaufe wird endlos wiederholt und immer wieder von oben nach unten abgearbeitet. Bei jedem Durchlauf, wird eine neue kleinere nach innen versetzte Kontur berechnet. Nur die zwei Abbruchbedingungen führen aus dieser Endlosschlaufe heraus. Die eine tritt ein, wenn die Kontur zu einem einzigen Punkt zusammengeschrumpft ist, die andere, wenn die gesuchte Kontur gefunden wurde. Die Schlaufe lässt sich in drei Bereiche unterteilen.

Im ersten Teil wird die Kontur aufbereitet. Unnütze Eckpunkte mit Status 0 oder Eckpunkte, die übereinander liegen werden entfernt. Zudem werden fehlerhafte Konturen mit Eckpunkten vom Status 2 repariert. Alles, was im ersten Teil durchgeführt wird, war leider nicht in der Anleitung

[12] erwähnt. Trotzdem ist dieser Teil wichtig. Ohne ihn würde der restliche Algorithmus nicht funktionieren.

Im zweiten Teil wird durch jeden Eckpunkt iteriert und für jeden der kritischen Punkt berechnet. Gesucht wird der entscheidende kritische Punkt mit dem kürzesten Abstand. Bei diesem Eckpunkt wird als erstes ein Ereignis eintreffen.

Im dritten Teil wird aus dem im zweiten Teil gefundenen kritischem Punkt eine neue Zwischenkontur berechnet. Falls der kritische Punkt durch einen konvexen Eckpunkt entstand, wird die neue Zwischenkontur mit einem Edge Event berechnet. Falls der kritische Punkt durch einen konkaven Eckpunkt entstand, wird die neue Zwischenkontur mit einem Edge, Split oder Merge Event berechnet.

4.7.13 Nachteil der Strahlkompensation

Die Strahlkompensation war mit Abstand die grösste Knacknuss bei der ganzen Programmierung. Da der Algorithmus an sich schon schwierig war, gestaltete sich die Implementierung sehr schwierig. Um das Ganze überhaupt implementieren zu können, mussten Kompromisse zu Lasten der Effizienz eingegangen werden. Der Code ist also nicht auf Geschwindigkeit optimiert. Die Berechnungszeit steigt mit der Anzahl Eckpunkte exponentiell an. Bei komplexen Konturen wird deshalb von der Strahlkompensation abgeraten.

4.8 Benutzeroberfläche

Die Benutzeroberfläche macht das Programm übersichtlicher und benutzerfreundlicher. Zudem kann die Darstellung der vielen einzelnen Bahnlinien verbessert werden. Da es in Matlab die Möglichkeit gibt, Benutzeroberflächen zu erstellen, wurden diese auch genutzt.

4.8.1 Aufbau der Benutzeroberfläche

In Abbildung 4.21 ist die Benutzeroberfläche des CAM Programms abgebildet. Sie lässt sich in zwei Bereiche unterteilen. Im linken Bereich können die Parametereingaben zur Berechnung der Laserbahnen getätigten werden. Im rechten Bereich ist eine Grafik. Diese dient zur Visualisierung der Stl-Datei, der berechneten Konturen und der Laserbahnen.

Die Felder, bei denen keine Eingaben getätigten werden können, sind stets ausgegraut. Damit lassen sich unnötige und falsche Eingaben von Beginn an vermeiden.

Bei der Programmierung wurde versucht die Benutzeroberfläche so intuitiv und selbsterklärend wie möglich zu gestalten. In diesem Kapitel wird deshalb nicht auf jedes einzelne Eingabefeld eingegangen.

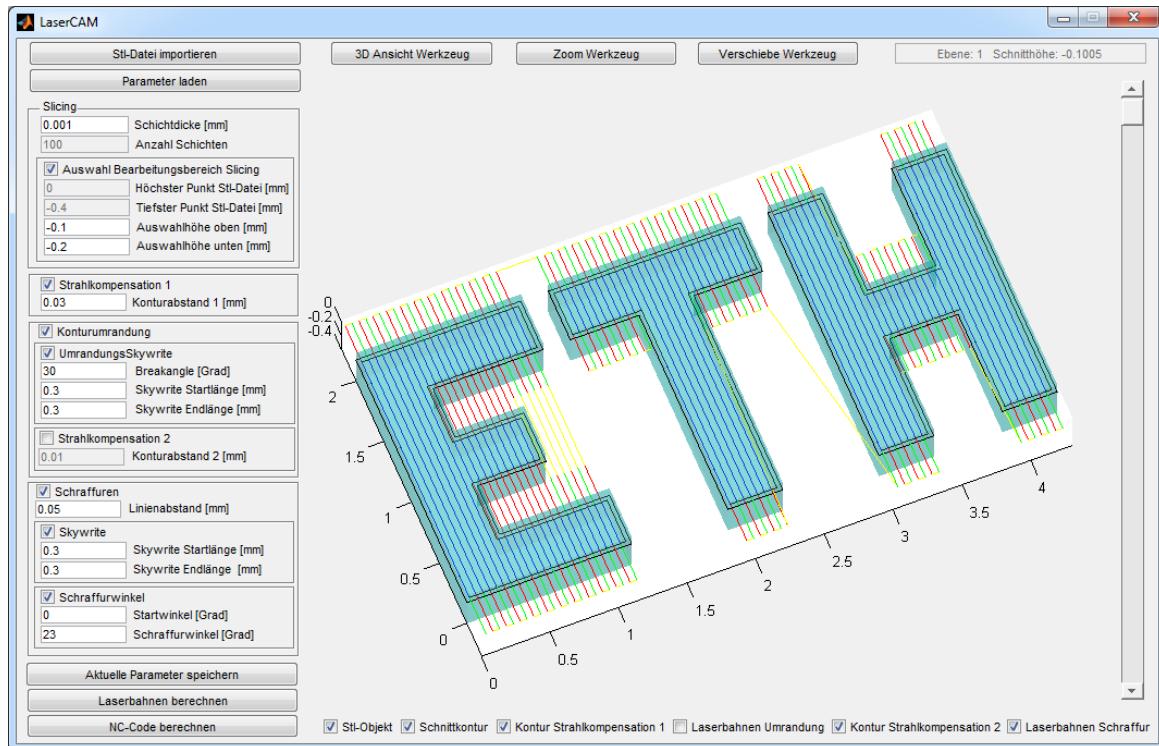


Abbildung 4.21: Benutzeroberfläche des CAM Programms

4.8.2 Parametereingabe zur Bahnberechnung

In der linken Spalte der Benutzeroberfläche (siehe Abbildung 4.21) in der die Eingabe der Parameter erfolgt arbeitet man sich von oben nach unten durch. Jeder umrahmte Block stellt einen Teilschritt in der Berechnung dar. Die meisten Blöcke können gezielt durch Setzen eines Häkchens über dem entsprechenden Block ausgewählt werden. Blöcke, die durch kein Häkchen angewählt sind, werden ausgegraut und fliessen nicht in die Berechnung ein. Dadurch kann Rechenzeit gespart werden.

Der Block „Auswahl Bearbeitungsbereich Slicing“ ist optional. Mit dieser Auswahl, werden nur die Schnittkonturen in einem bestimmten Höhenbereich berechnet und nicht alle Schnittkonturen über die gesamte Höhe der Stl-Datei. Die Eingabe des neuen Auswahlbereichs erfolgt über die Felder „Auswahlhöhe oben [mm]“ und „Auswahlhöhe unten [mm]“ und muss innerhalb den Werten „Höchster Punkt Stl-Datei [mm]“ und „Tiefster Punkt Stl-Datei [mm]“ liegen.

Mit der Schaltfläche „Parameter laden“ können gespeicherte Einstellungen geladen werden. Dadurch müssen die Parameter nicht jedes Mal von neuem eingegeben werden.

Über die Schaltfläche „Aktuelle Parameter speichern“ können die Einstellungen in einer Textdatei gespeichert werden.

Mit der Schaltfläche „Laserbahnen berechnen“ wird die Berechnung gestartet. Die berechneten Geometrien werden fortlaufend in der Grafik im rechten Bereich der Benutzeroberfläche dargestellt.

4.8.3 Erstellen des NC-Codes

Durch Anklicken der untersten Schaltfläche „NC-Code berechnen“, (siehe Abbildung 4.21) öffnet sich das in Abbildung 4.22 dargestellte Fenster. In diesem können die einzelnen NC-Befehle an die Maschine angepasst werden, auf welcher der NC-Code ausgeführt wird.

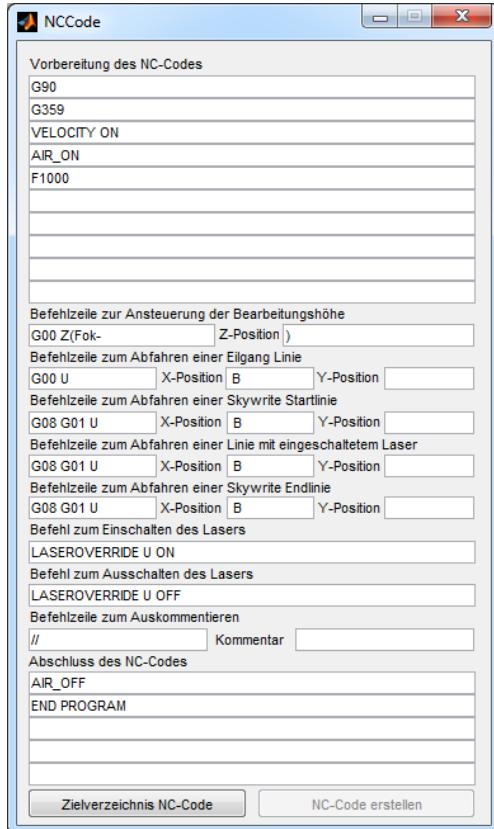


Abbildung 4.22: Anpassung der NC-Befehle

Der Befehl zur Ansteuerung der Bearbeitungshöhe gliedert sich in drei Felder und wird zu einer NC-Codezeile zusammengesetzt. Die Eingabe vor und nach dem Feld „Z-Position“ kann frei angepasst werden. An der Stelle „Z-Position“ wird bei der Erstellung des NC-Codes automatisch die richtige Bearbeitungshöhe eingefügt. Diese beginnt bei null und wird von Schicht zu Schicht um eine Schichtdicke erhöht.

Die NC-Codezeilen der verschiedenen Bahnlinien setzen sich jeweils aus fünf Feldern zusammen. Die Eingaben in den weissen Feldern können frei angepasst werden. Die X-Position und Y-Position, entspricht der Zielposition der entsprechenden Bahnlinie und richtet sich an den Koordinaten der Stl-Datei.

Nachdem ein Zielverzeichnis für den NC-Code ausgewählt wurde, kann dieser erstellt werden. Der NC-Code wird als Textdatei ausgegeben.

4.8.4 Einstellungsmöglichkeiten zur grafischen Darstellung

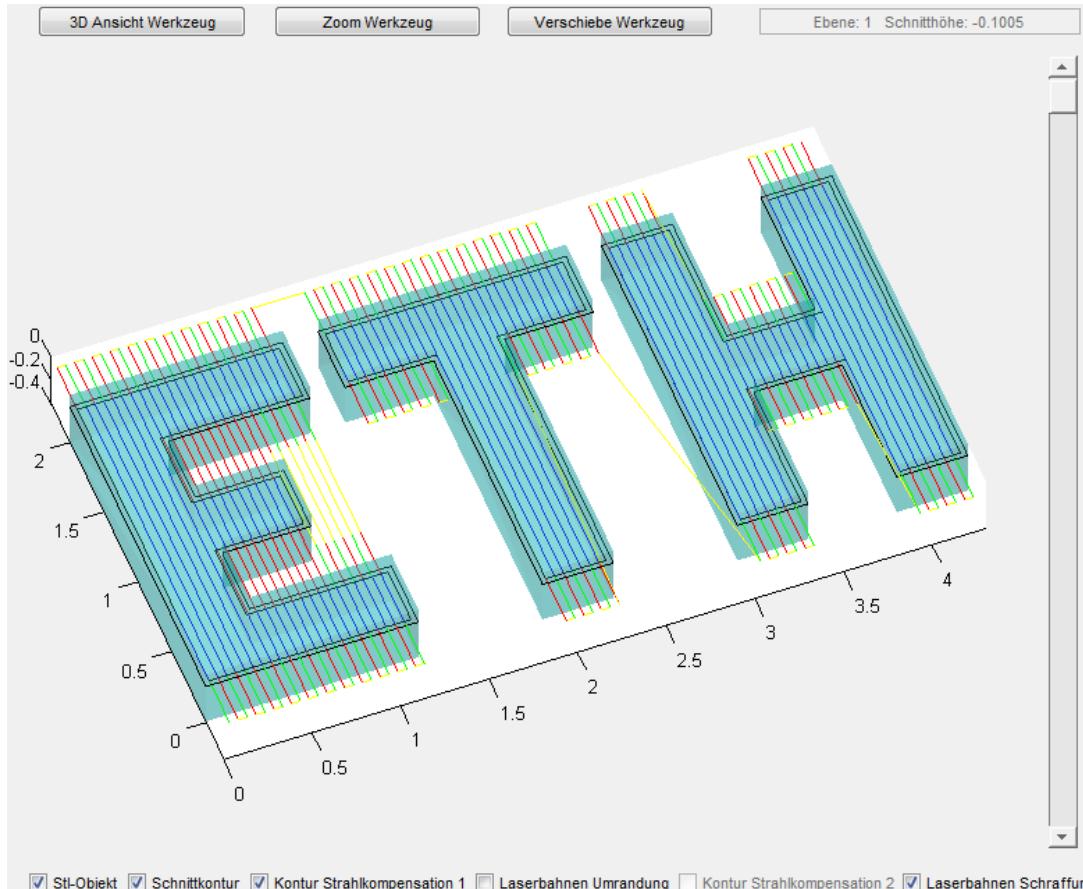


Abbildung 4.23: Grafische Darstellung der Stl-Datei, Konturen und Laserbahnen

Der rechte Bereich der Benutzeroberfläche mit der grafischen Darstellung ist in Abbildung 4.23 ersichtlich. Die Stl-Datei ist dabei hellblau und die Konturen als schwarze Linien visualisiert. Die Skywrite Startlinien sind als grüne, die Skywrite Endlinien als rote, die Eilgänge als gelbe und die Bahnen mit eingeschaltetem Laser als blaue Linien dargestellt. Damit die Grafik übersichtlich bleibt, können die einzelnen Darstellungen gezielt ein- oder ausgeblendet werden. Dazu gibt es am unteren Rand sechs Auswahlkästchen.

Mit dem Slider am rechten Rand der Benutzeroberfläche, kann man auswählen von welcher Schnitthöhe die Konturen und Laserbahnen angezeigt werden sollen. In der oberen, rechten Ecke wird ausgeschrieben, in welcher Schnittebene und Schnitthöhe sich die dargestellten Konturen und Laserbahnen befinden.

Am oberen Rand der Benutzeroberfläche, sind drei Schalflächen angeordnet. Mit diesen kann die Ansicht der Grafik verändert werden. Ist das „3D Ansicht Werkzeug“ selektiert, kann die Grafik durch verschieben des Mauszeigers mit gedrückter, linker Maustaste rotiert werden. Mit dem „Zoom Werkzeug“ kann durch Scrollen in die Grafik hinein- oder hinausgezoomt werden. Ist das „Verschiebe Werkzeug“ selektiert, kann durch Verschieben des Mauszeigers mit gedrückter, linker Maustaste, die Grafik verschoben werden.

5

Ergebnisse und Diskussion

Zur Validierung des programmierten CAMs wurden mehrere Tests durchgeführt. Zur Diskussion werden in diesen Kapitel zwei gefertigte Werkstücke genauer betrachtet. Das Ausgangswerkstück war in beiden Fällen ein quadratisches Hartmetallplättchen mit einer Fläche von zehn mal zehn Millimetern und einer Dicke von zwei Millimetern.

5.1 Werkstück mit ETH Logo

Bei diesem Versuch wurde durch die Laserbearbeitung ein ETH Logo abgetragen. Eine Aufnahme des Resultats ist in Abbildung 5.1 abgebildet. Das Logo ist 8.6 Millimeter lang und 4 Millimeter hoch. Es wurde in 100 Schichten abgetragen. Die erreichte Tiefe beträgt 91 Mikrometer.

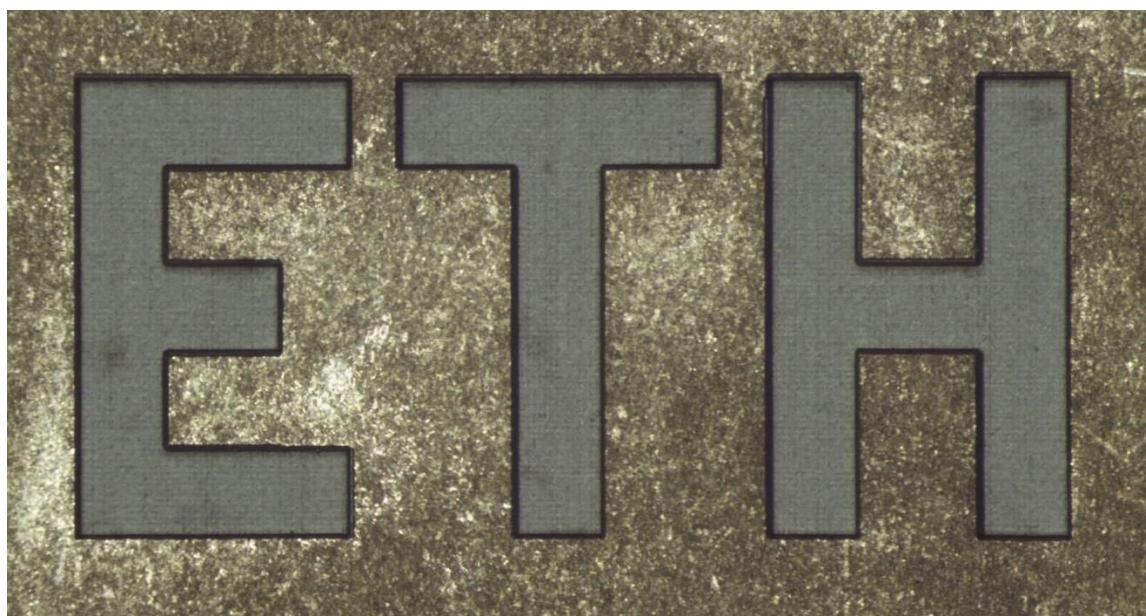


Abbildung 5.1: Mikroskopaufnahme des ETH Logos

5.1.1 Die gewählten Bearbeitungsparameter für das ETH Logo

Tabelle 5.1 gibt einen Überblick über die Parameter, die bei der Laserbearbeitung des ETH Logos verwendet wurden. Diese lehnen sich an jene der Semesterarbeit von Pascal Gilgen [19] an. Das Laser on und off Delay wurde spezifisch für die CNC-Steuerung A3200 von Aerotech und den daran angeschlossenen Scankopf hurrySCAN II 14 bestimmt. Die Skywritelängen von 500 Mikrometern sind konservativ gewählt. Zur Verkürzung der Bearbeitungszeit können diese reduziert werden.

Scangeschwindigkeit	1000mm/s
Linienabstand	5 μm
Schichtdicke	0.5 μm
Variabler Schraffurwinkel	90°
Skywrite Startlänge	500 μs
Skywrite Endlänge	500 μs
Strahlkompensation	15 μm
Laser on Delay	190 μs
Laser off Delay	210 μs
Impulsfrequenz des Lasers	672 kHz
Durchschnittliche Laserleistung	2.6 W
Pulsenergie	3.9 μJ

Tabelle 5.1: Verwendete Parameter für das Fertigen des ETH Logos

Die ersten sieben Parameter von Tabelle 5.1, waren ausreichend, um mit dem CAM Programm den NC-Code zu berechnen. Dieser war 46.7 Megabyte gross und beinhaltete 1'457'809 Zeilen.

5.1.2 Der Materialabtrag beim ETH Logo

Abbildung 5.2 links zeigt eine topografische Oberflächenaufnahme des ETH Logo. Dargestellt ist eine Vergrösserung des unteren Endes des Buchstabens T. Orange visualisiert ist die Oberfläche des unbearbeiteten Hartmetalls. Die durch den Laserabtrag erhaltene Oberfläche ist blau dargestellt.

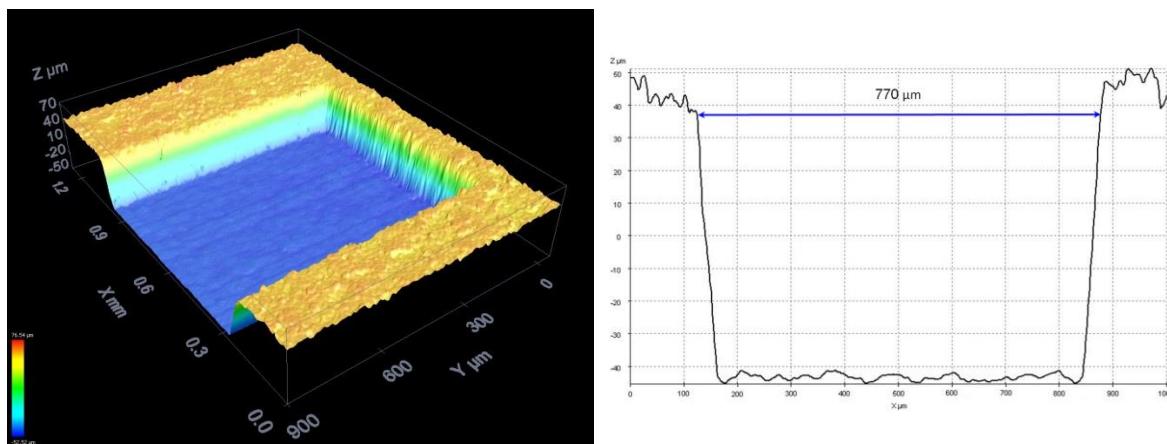


Abbildung 5.2: Detailaufnahmen des abgetragenen ETH Logos

Durch die Laserbearbeitung hat sich die Oberflächenrauheit verringert. Die unbearbeitete Oberfläche hat einen Ra-Wert von $1.6 \mu\text{m}$. Der Ra-Wert auf dem Grund des ETH Logos beträgt $0.9 \mu\text{m}$. Durch die Laserbearbeitung wurde nicht nur Material abgetragen, sondern auch die Oberfläche geglättet.

In Abbildung 5.2 rechts ist das Profil entlang der X-Achse durch den unteren Teil des Buchstabens T des ETH Logos abgebildet. Die konischen Abtragsflanken sind typisch für die 2.5D Laserbearbeitung. Der senkrecht über den Abtragsflanken einfallende Laserstrahl wird an diesen reflektiert und trägt unterhalb der Flanken etwas mehr Material ab. An diesen Stellen ist ein erhöhter Materialabtrag sichtbar.

Die Breite der abgetragenen Tasche sollte eigentlich 800 Mikrometer sein. Wie in Abbildung 5.2 rechts ersichtlich, ist die durch die Laserbearbeitung erhaltene Breite nur 770 Mikrometer breit. Diese ist also 30 Mikrometer zu klein. Ohne die Strahlkompensation, von je 15 Mikrometern auf beiden Seiten hätte die Tasche genau die richtigen Dimensionen.

5.2 Werkstück mit abgetragenem Laserwarnzeichen

In diesem Versuch wurde das in Abbildung 5.3 abgebildete Laserwarnzeichen abgetragen. Jede Kantenlänge des Warndreiecks ist 9.4 Millimeter lang. Der Abtrag erfolgte in 100 Schichten. Dadurch wurde eine Tiefe von 91 Mikrometer erreicht.

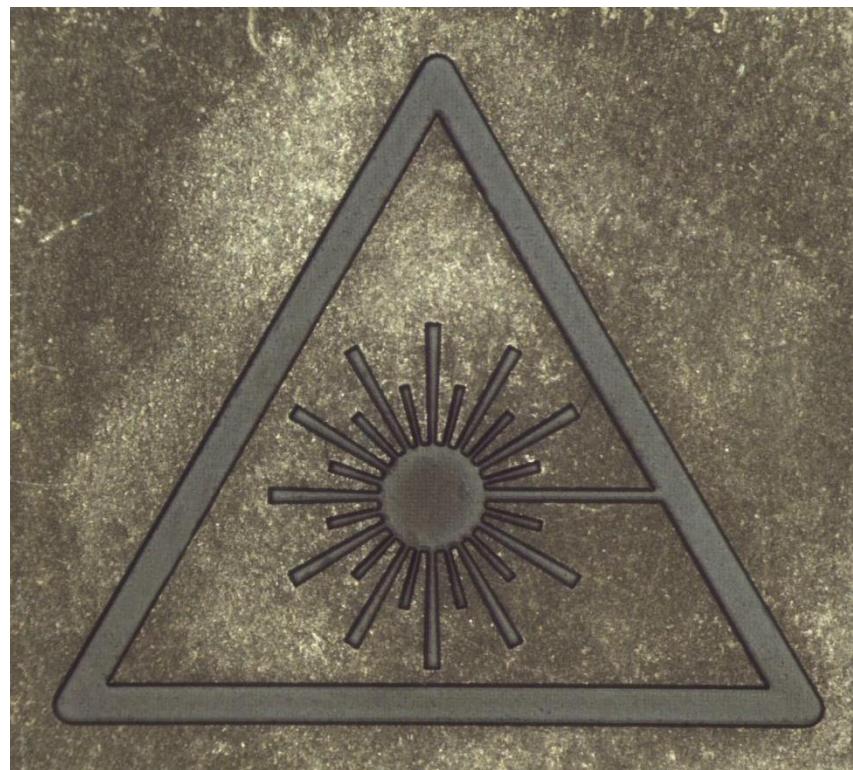


Abbildung 5.3: Mikroskopaufnahme des gefertigten Laserwarnzeichens

Die komplexe Geometrie stellt eine erhöhte Herausforderung dar. Deshalb wurden die Scangeschwindigkeit, die Impulsfrequenz und die Laserleistung bei diesem Versuch halbiert.

5.2.1 Die Bearbeitungsparameter für das Laserwarnzeichen

In Tabelle 5.2 sind die Parameter, die bei der Fertigung des Laserwarnzeichens verwendet wurden aufgelistet. Bei diesem Versuch wurde ein Schraffurwinkel von 23 Grad gewählt. Dadurch gibt es keine Überlagerung der Schraffurrichtungen zwischen den einzelnen Schichten.

Scangeschwindigkeit	500 mm/s
Linienabstand	5 μm
Schichtdicke	0.5 μm
Variabler Schraffurwinkel	23°
Skywrite Startlänge	500 μs
Skywrite Endlänge	500 μs
Strahlkompensation	15 μm
Laser on Delay	190 μs
Laser off Delay	210 μs
Impulsfrequenz des Lasers	336 kHz
Durchschnittliche Laserleistung	1.3 W
Pulsenergie	3.9 μJ

Tabelle 5.2: Verwendete Parameter zur Fertigung des Laserwarnzeichens

Durch die komplexe Geometrie wurde der NC-Code länger. Er beinhaltet 3'490'611 Zeilen und war 113MB gros. Die Berechnung des Codes dauerte rund drei Minuten.

5.2.2 Oberflächenqualität beim Laserwarnzeichen

Abbildung 5.4 links zeigt eine Vergrösserung des unteren Randes des Laserwarnzeichens. Der Ra-Wert der unbehandelten, orangen Oberfläche beträgt 1.6 μm . Derjenige auf dem blau dargestellten Grund 0.6 μm . Mit dem ändernden Schraffurwinkel von 23 Grad konnte die Rillenbildung verringert und die Oberflächenqualität gesteigert werden. Das ist auch in Abbildung 5.4 links zu erkennen.

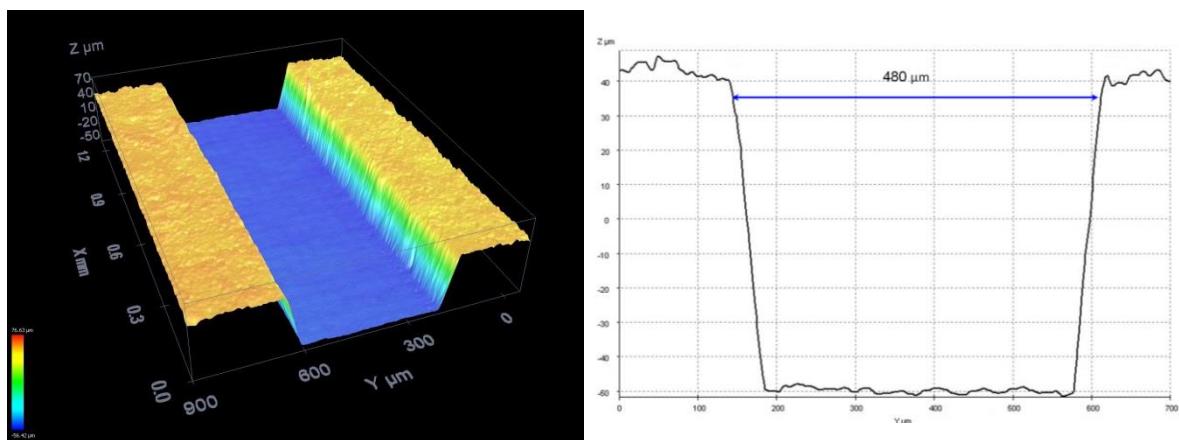


Abbildung 5.4: Detailaufnahmen des Laserwarnzeichens

In Abbildung 5.4 rechts ist das Profil entlang der Y-Achse durch den unteren Rand des Laserwarnzeichens dargestellt.

Die Breite der Tasche sollte eigentlich 500 Mikrometer sein. Wie in Abbildung 5.4 rechts ersichtlich, ist die gefertigte Breite nur 480 Mikrometer. Die resultierende Breite ist also 20 Mikrometer zu klein. Auch hier ist der gewählte Konturabstand bei der Strahlkompensation zu gross gewählt.

Welches die optimalen Parameter bei der Stahlkompensation sind, wäre in weiteren Versuchen zu bestimmen.

6

Schlussfolgerung und Ausblick

Die Erstellung eines CAM-Systems für die 2D und 2.5D Laserbearbeitung konnte in dieser Arbeit erfüllt werden. Das Programm ist ein funktionsfähiges, abgeschlossenes Projekt, das die wichtigsten Funktionen für die Laserbearbeitung zur Verfügung stellt.

Das Programm ist in erster Linie auf Funktionalität und Zuverlässigkeit ausgelegt. Bei den Berechnungen, die durchgeführt wurden, hat alles gut funktioniert. Es wurde versucht das Programm so robust wie möglich zu schreiben. Wie bei jedem Programm kann es trotzdem sein, dass Fehler enthalten sind. Diese werden wahrscheinlich erst bei intensiver Benutzung auftreten.

Da die Funktionalität des Programms oberste Priorität hatte, war die Effizienz zweitrangig. Diese kann noch optimiert werden. Durch straffen des Codes oder effizientere Algorithmen könnte diese verbessert werden. Durch Aufteilen der Aufgaben auf mehrere Prozessorkerne kann die Berechnungszeiten verkürzt werden.

Die Prozessparameter für die Laserbearbeitung, die in dieser Arbeit verwendet wurden, bieten eine funktionierende Grundlage. Zur Verkürzung der Bearbeitungszeit und für eine verbesserte Werkstückqualität können diese noch optimiert werden. Durch die Benutzerfreundlichkeit des CAM Programms wird das Finden der optimalen Parameter und das Erstellen des NC-Codes sicher vereinfacht.

Durch den modularen Aufbau des CAM Programms kann dieses einfach an weitere Bedürfnisse und Prozesse angepasst werden. Hierfür steht der Programmcode jedem zur Verfügung.

Um das Programm für das 3D-Drucken zu verwenden, sind nicht viele Änderungen notwendig. Der Programmcode müsste nur so angepasst werden, dass die einzelnen Schichten nicht abgetragen, sondern aufgetragen werden.

Durch das Hinzufügen von Funktionen, die zusätzliche Drehachsen beherrschen, könnte das Werkstück zwischen den einzelnen Abtragsprozessen gedreht werden. Dadurch ist beispielsweise die Herstellung zylindrischer Werkstücke denkbar.

A

Anhang

A.1 Anleitung zum Ausführen des NC-Codes

In dieser Anleitung ist beschrieben, wie der NC-Code, der mit dem CAM Programm erstellt wurde, auf der CNC-Steuerung von Aerotech ausgeführt werden kann.

Zuerst wird der Motion Composer von Aerotech gestartet. Mit diesem Programm können die verschiedenen Achsen angesteuert werden. In Abbildung A.1 ist ein Screenshot der Benutzeroberfläche dargestellt.

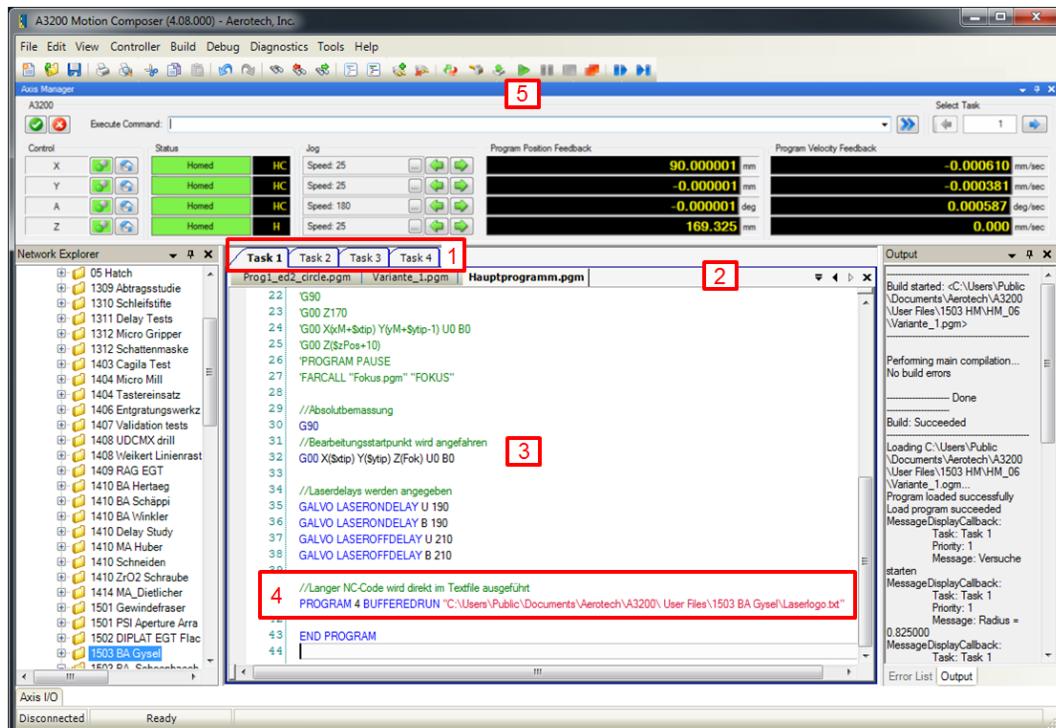


Abbildung A.1: Benutzeroberfläche des Motion Composer von Aerotech

A Anhang

Auf der Benutzeroberfläche gibt es vier Tasks. Diese sind in Abbildung A.1 rot umrandet und mit der Nummer 1 gekennzeichnet. Hier wird der Task 1 angewählt. Durch einen Rechtsklick in die leere Taskleiste (siehe Abbildung A.1 Nummer 2) kann ein neues leeres NC-Programm erstellt werden. In das Feld, das in Abbildung A.1 mit der Nummer 3 markiert ist, wird dann folgender Programmcode eingegeben:

```
DVAR $incr
DVAR $zPos
DVAR $xtip
DVAR $ytip

//TRUE, wenn Werkstück in A-Achse
Wsa = 0
//Werkstück Länge (ab HSK nach rechts)
WSl = 100
//Werkstück Höhe / Radius
WSh = 1
FARCALL "Safezone.pgm" "SAFE"

POSOFFSET CLEAR U B
YF25 ZF25

$zPos = WSh
$xtip = 30
$ytip = 0

,//Programmteil zur Antastung des Werkstücks
,G90
,G00 Z170
,G00 X(xM+$xtip) Y(yM+$ytip-1) U0 B0
,G00 Z($zPos+10)
,PROGRAM PAUSE
,FARCALL „Fokus.pgm“ „FOKUS“

//Absolutbemassung
G90
//Bearbeitungsstartpunkt wird angefahren
G00 X($xtip) Y($ytip) Z(Fok) U0 B0

//Laserdelays werden angegeben
GALVO LASERONDELAY U 190
GALVO LASERONDELAY B 190
GALVO LASEROFFDELAY U 210
GALVO LASEROFFDELAY B 210

//Langer NC-Code wird direkt im Textfile ausgeführt
PROGRAM 4 BUFFEREDRUN "C:\Users\Public\Documents\Aerotech\A3200\User
Files\1503 BA Gysel\Laserlogo.txt"
END PROGRAM
```

Dieser Programmcode ist zusätzlich auf der CD im Anhang gespeichert. Der Pfad, der auf die Textdatei verweist, die mit dem CAM Programm erstellt wurde, muss entsprechend angepasst werden. Dieser ist in Abbildung A.1 rot umrandet und mit der Nummer 4 gekennzeichnet. Danach kann das NC-Programm mit der Startschaltfläche (siehe Abbildung A.1 Nummer 5) ausgeführt werden.

A.2 Daten auf der beigelegten CD

Auf der beigelegten CD sind diverse Daten zu dieser Arbeit gespeichert. Hier ein kurzer Überblick

A.2.1 Sourcecode CAM Programm

Im Ordner „Matlab LaserCAM“ ist sämtlicher Matlab Code enthalten, aus dem das CAM Programm aufgebaut ist.

A.2.2 Kompiliertes CAM Programm

Eine mit Matlab kompilierte Programmversion des CAM Programms ist im Ordner „LaserCAM“ enthalten. Diese Version kann ausgeführt werden ohne das Matlab installiert ist.

A.2.3 Stl-Dateien

Im Ordner „Stl-Dateien“ sind verschiedene Modellgeometrien, die im Rahmen dieser Arbeit verwendet wurden. Diese können mit dem CAM Programm geöffnet werden.

A.2.4 NC-Code

Im Ordner „NC-Code“ befinden sich mehrere Textdateien. Die eine beinhaltet den NC-Code mit dem das ETH Logo gefertigt wurde. Der NC-Code mit dem das Laserwarnzeichen erstellt wurde ist ebenfalls enthalten. Zudem ist in diesem Ordner der Programmcode mit dem von der CNC-Steuerung von Aerotech der NC-Code in den Textfiles ausgeführt werden kann.

A.2.5 Grafiken

Zur Verständlichkeit dieser Arbeit habe ich viele Grafiken erstellt. Diese sind alle im Ordner „Grafiken“ enthalten und dürfen für weitere Zwecke verwendet werden.

Literaturverzeichnis

- [1] Wegener, K., Einführung NC-Technik, Skript zur Vorlesung Fertigungstechnik I, Seite 2, 2010
- [2] Renk, K., F., Basics of Laser Physics for Students of Science and Engineering, Springer, 2012
- [3] Eberle, G., Dold, C., Wegener, K., Laser fabrication of diamond micro cutting tool related geometries using a high-numerical aperture micro scanning system, Institute of Machine Tools and Manufacturing (IWF), 2015
- [4] Lehner, L., Herstellung von Werkzeugen für hohe Vorschübe mittels Laserbearbeitung, Bachelorarbeit, 2014
- [5] Sintec Optronics Technology Pte Ltd, Laser Marking Heads (Laser Scanners, Optical Scanners, Scan Heads), <http://www.sintecoptronics.com/markinghead.asp>, [Abgerufen: 26.5.2015]
- [6] SCAPS GmbH, SAM Light Benutzerhandbuch, 2013
- [7] Ultimaker, <http://wiki.ultimaker.com/Cura>, [Abgerufen: 26.5.2015]
- [8] Hon Wah, W., Introduction to STL format, 1999
- [9] Jaeggi, B., Neuenschwander, B., Meier, T., Zimmermann, M., Hennig, G., High throughput laser micro machining on a rotating cylinder with ultra short pulses at highest precision, Bern University of Applied Sciences, 2013
- [10] Dietlicher, M., Bahnoptimierung für synchronisierte Laser- und CNC-Achsen, Masterarbeit, 2015
- [11] Dörig, Ch., Bearbeitung von Hartmetall- und CVD-Fräsern mittels Laser für die Zerspanung von CFK, Bachelorarbeit, 2013
- [12] Felkel, P., Obdržálek, S., Straight Skeleton Implementation, 1998
- [13] Time-Bandwidth, DUETTO User Manual, 2010
- [14] Aerotech, <http://www.Aerotech.com/product-catalog/motion-controllers/a3200.aspx>, [Abgerufen: 15.5.2015]

- [15] ISO 6983-1, Automation systems and integration – Numerical control of machines – Program format and definitions of address words, Internationl Organization for Standardization, 2009
- [16] The MathWorks Inc., <http://ch.mathworks.com/help>, [Abgerufen: 26.5.2015]
- [17] The MathWorks Inc., <http://ch.mathworks.com/matlabcentral/fileexchange/6678-stlread>, [Abgerufen: 13.5.2015]
- [18] Leica, <http://www.leica-microsystems.com>, [Abgerufen: 13.5.2015]
- [19] Gilgen, P., Laserbearbeitung von PKD und Hartmetall: Erzeugung einer axialen Nut in einem Zwei-Komponenten-Zylinderrohling, Seite 39, 2013

Plagiatserklärung

Ich erkläre mit meiner Unterschrift, das Merkblatt Plagiat zur Kenntnis genommen, die vorliegende Arbeit selbständig verfasst und die im betroffenen Fachgebiet üblichen Zitievorschriften eingehalten zu haben.

Merkblatt Plagiat: http://www.ethz.ch/students/semester/plagiarism_s_de.pdf

Ort, Datum

Unterschrift