

Beitrag zur voxelbasierten Simulation des fünfachsigen NC-FräSENS

vorgelegt vom
Master of Engineering
Zengxuan Hou
aus Shaanxi, VR China

von der Fakultät V - Verkehrs- und Maschinensysteme
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr.-Ing. H. Pucher
Berichter:	Prof. Dr.-Ing. F.-L. Krause
Berichter:	Prof. Dr.-Ing. E. Uhlmann

Tag der wissenschaftlichen Aussprache: 14. Juli 2003

Berlin 2003

D 83

Vorwort des Herausgebers

Die Simulationstechnik wird als Hilfsmittel zur Beherrschung der drei- und fünfachsigen NC-Fräsbearbeitung eingesetzt. Gewöhnlich wird die Aktualisierung des simulierten Werkstücks entweder im Bildraum des Werkstücks oder geometrisch im Objektraum durchgeführt. Die vollständige Aktualisierung des simulierten Werkstücks fordert jedoch nur eine geometrische Lösung.

Im vorliegenden Forschungsbericht wird der Ansatz der voxelbasierten fünfachsigen NC-Simulation entwickelt. Die Grundlage für den Aufbau des Simulationssystems ist der am IPK Berlin entwickelte 3D-Kernmodellierer des Virtual Clay Modelling Systems. Dieser ermöglicht die Modellierung und die Aktualisierung von komplizierten Werkstücken und Zerspanungsvolumen in der drei- und fünfachsigen NC-Simulation auf Rechenanlagen.

Das simulierte Werkstück wird rechnerintern diskret mit einem Voxelmodell dargestellt. Die dadurch eingeführten NC-Simulationsfehler werden unter Berücksichtigung des eingesetzten Fräasers kontrolliert. Eine voxelbasierte Methode zur Gestaltung von Zerspanungsvolumen der diversen Werkzeuge wie zylindrische Fräser, Torus-, zylindrische Gesenk-, Kugelkopf-, Fass-, Kegel- und kegelige Gesenkfräser wurde entwickelt. Der Vorteil dieser Methode liegt darin, dass aufgrund der diskreten Datenstruktur des Voxelmodells und der damit ermöglichten Abbildung von beliebigen Formen keinerlei Einschränkungen zur Modellierung der Zerspanungsvolumen der diversen Fräserarten gegeben sind.

Die Materialverletzungen und verbliebenen Restmaterialien auf dem simulierten Werkstück können im 3D-Raum zur Bewertung der Qualität der NC-Programme ermittelt werden. Eine effektive Optimierung der NC-Programme in der Fertigungsvorbereitung kann durchgeführt werden. Zur Beschleunigung der Werkstückaktualisierung wurden ausgehend von den Eigenschaften des Voxelmodells die entsprechenden Maßnahmen getroffen.

Mit der vorgestellten Methode wird ein wichtiger Beitrag zur drei- und fünfachsigen Simulation des NC-Fräsens geleistet. Damit können NC-Programme für die drei- und fünfachsige NC-Fräsbearbeitung verifiziert, optimiert und eine hohe Prozessqualität vorab sichergestellt werden.

Vorwort des Autors

Die vorliegende Arbeit entstand während meiner Tätigkeit als Gastwissenschaftler am Institut für Werkzeugmaschinen und Fabrikbetrieb (IWF) der Technischen Universität Berlin und am Fraunhofer Institut für Produktionsanlagen und Konstruktionstechnik (IPK).

Herrn Prof. Dr.-Ing. F.-L. Krause, dem Leiter des Fachgebiets Industrielle Informationstechnik des IWF und des Bereichs Virtuelle Produktentwicklung des IPK, danke ich für die Gelegenheit zur Promotion und für seine stetige Unterstützung. Herrn Prof. Dr.-Ing. E. Uhlmann, dem Leiter des IPK, danke ich für das dieser Arbeit entgegengebrachte Interesse und die Übernahme der Berichterstattung. Gleichfalls bedanke ich mich bei Prof. Dr.-Ing. H. Pucher für die Übernahme des Vorsitzes im Promotionsausschuss und das von ihm gezeigte Interesse an meiner Arbeit.

Herzlich bedanken möchte ich mich beim Auslandsamt der Technischen Universität Berlin für die großzügige Förderung der Arbeit.

Weiterhin danke ich allen Mitarbeiterinnen und Mitarbeitern der beiden Institute, IWF und IPK, insbesondere den Kollegen und studentischen Mitarbeitern der Abteilung Produktmodellierung, für die vielfältige Hilfsbereitschaft.

Mein besonderer Dank gilt Herrn Dipl.-Ing. U. Rothenburg, dem Leiter der Abteilung Produktmodellierung, und Herrn Dipl.-Ing. Th. Wöhler, Frau Dipl.-Math. N. Gross, Herrn Dipl.-Ing. J. Neumann, Herrn Dipl.-Inform. A. Striepe, Herrn Dipl.-Ing. A. Romahn, Herrn H.-J. Greinert, für ihre stetige und wertvolle Unterstützung der Arbeit. Herrn Dipl.-Ing. G. Gross und Herrn Dipl.-Math. R. Eisenberg danke ich herzlich für die freundliche und ausgezeichnete Unterstützung in der letzten Phase der Arbeit.

Besonders möchte ich meiner Frau Yanfang Cao und meinem Sohn Yu Hou, die mit viel Geduld und Opferbereitschaft das Entstehen dieser Arbeit ermöglicht haben, herzlich danken.

Beitrag zur voxelbasierten Simulation des fünffachsigen NC-FräSENS

Inhaltsverzeichnis	Seite
Inhaltsverzeichnis.....	1
0 Verwendete Abkürzungen	5
1 Einleitung und Zielsetzung	7
2 Stand der Technik	10
2.1 Allgemeine Betrachtung	10
2.2 Volumenorientierte Repräsentationen	11
2.2.1 Direkte Volumenrepräsentationen	11
2.2.2 Räumliche partitionierte Repräsentationen	13
2.2.3 Zusammenfassung	18
2.3 Zerspanungsvolumen	18
2.4 Fehlerbetrachtung in der NC-Simulation	21
2.5 Visualisierung in der Werkstückaktualisierung	22
2.5.1 Direktes Verfahren	22
2.5.2 Indirektes Verfahren	24
2.6 NC-Simulationssysteme	24
2.7 Anforderungen an die Werkstückaktualisierung	27
3 Gestaltung des Voxelmodells zur NC-Simulation unter Beschränkung der Simulationsfehler	29
3.1 Virtual Clay Modelling System - VCM-System	29
3.2 Das Voxelmodell im VCM-System	31
3.2.1 Komprimierte Voxeldarstellung durch Stixel-Repräsentation	31
3.2.2 Die Mischoperationen	32

3.2.3	Datenstruktur	33
3.3	Marching-Cubes-Verfahren zur Visualisierung des Voxelmodells.....	36
3.4	Fehlerbegrenzung in der voxelbasierten fünfachsigem NC-Simulation	37
3.4.1	Begrenzung des Flächenfehlers in der voxelbasierten Geometrie- darstellung.....	37
3.4.2	Begrenzung des Unterschnittfehlers in der voxelbasierten fünfachsigem NC- Simulation	39
3.4.3	Begrenzung des Restmaterialfehlers in der voxelbasierten fünfachsigem NC-Simulation.....	41
3.4.4	Zusammenfassung	45
3.5	Erstellung eines Voxelmaterialmodells mit dem VCM-System	47
4	Voxelbasierte Gestaltung des Zerspanungsvolumens in der NC- Simulation.....	50
4.1	Konzept zur Gestaltung des Zerspanungsvolumens in der voxelbasierten NC-Simulation	50
4.1.1	Leitgedanken zur Konzeption	50
4.1.2	Klassifizierung der Grundgeometrie zur Darstellung von diversen Fräsern	52
4.1.3	Methode zur Darstellung der Oberflächen des Grundkörpers und dessen Bewegungsmodells	54
4.1.4	Rechnerinterne Informationsstruktur der Fräser	55
4.2	Gestaltung des Bewegungsvolumens der diversen Grundkörper in der fünfachsigem NC-Simulation	57
4.2.1	Gestaltung des Bewegungsvolumens einer Kugel	57
4.2.2	Gestaltung des Bewegungsvolumens eines Zylinders	59
4.2.3	Gestaltung des Bewegungsvolumens eines Torus	61
4.2.4	Gestaltung des Bewegungsvolumens eines Kegels.....	64
4.2.5	Gestaltung des Bewegungsvolumens eines Fasses.....	66
4.2.6	Gestaltung der Oberflächen der diversen Bewegungsvolumen mit Dreiecksnetzen	69
4.3	Voxelbasierte Diskretisierung der Grundkörper und Bewegungsvolumen.	71
4.3.1	Diskretisierung des Grundkörpers	71
4.3.2	Diskretisierung der dreiecksbasierten Bewegungsvolumen	71
4.4	Gestaltung des voxelbasierten Zerspanungsvolumens	76
5	Aktualisierung des simulierten Werkstücks	78

5.1	Allgemeiner Ablauf der Werkstückaktualisierung	78
5.2	Lokalisierung	82
5.3	Variable Visualisierungen des simulierten Werkstücks.....	83
5.3.1	Ausnutzung der Stixelrepräsentationsform bei der Oberflächenextraktion	83
5.3.2	Segmentierte Visualisierung.....	84
5.3.3	Darstellungsverfahren durch variable Auflösung	85
5.3.4	Gesteuerte Visualisierung.....	86
5.4	Konzept der Parallelisierung.....	87
6	Methoden zur Erkennung der Materialverletzungen und Ermittlung der verbliebenen Restmaterialien	92
6.1	Allgemeine Vorgehensweise bei der voxelbasierten Verifikation des NC- Programms	92
6.2	Erkennung von Materialverletzungen	95
6.3	Ermittlung von verbliebenen Restmaterialien	97
6.4	Zusammenfassung	97
7	Systemrealisierung und Anwendungsbeispiel	99
7.1	Funktionsbeschreibung	99
7.2	Benutzungsoberfläche	101
7.2.1	Eingesetztes Graphiksystem.....	101
7.2.2	Der realisierte Benutzerdialog.....	102
7.2.3	Graphische Darstellungsmöglichkeiten	105
7.3	Simulationssteuerung.....	106
7.4	Anwendungsbeispiel	109
7.4.1	Beschreibung eines Verdichterlaufrades.....	109
7.4.2	NC-Programme eines Verdichterlaufrades	111
7.4.3	Fünfsichtige NC-Simulation eines Verdichterlaufrades.....	113
8	Zusammenfassung.....	117
9	Anhang A Ermittlung des maximalen Unterschnittfehlers bei der voxelbasierten Repräsentation	120
9.1	Überlappung des Werkzeugs mit zwei Eckpunkten	121

9.2	Überlappung des Werkzeugs mit drei Eckpunkten, aber nicht durch zwei Endpunkte einer Kante des Würfels	121
9.3	Überlappung des Werkzeugs mit drei Eckpunkten, aber zwei davon gehören zu einer Kante des Würfels	122
10	Anhang B Beispiele von Volumenspurmodellen	124
11	Anhang C Beispiele von Simulationsmodell	129
12	Literatur	133

0 Verwendete Abkürzungen

AutoCAD	CAD-System der Fa. AutoDesk
B-Rep	Boundary Representation
B-Spline	Basis-Spline
C	problemorientierte höhere Programmiersprache
C++	objektorientierte hybride Programmiersprache (vereinigt C mit objektorientierter Programmierung)
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CLDATA	Cutter Location Data, Werkzeugwegpositionsdaten nach DIN 66215
CNC	Computerized Numerical Control
CSG	Constructive Solid Geometry
Dexel	Depth Element
DIN	Deutsches Institut für Normung e.V.
DV	Datenverarbeitung
FHG	Fraunhofer-Gesellschaft zu Förderung der angewandten Forschung e.V.
IPK	Institut für Produktionsanlagen und Konstruktionstechnik der Fraunhofer-Gesellschaft
Open Inventor	objektorientierte 3D-Graphikbibliothek auf der Basis von Open GL mit dazugehörigem Austauschformat
IRIX	eine UNIX-Variante für Silicon Graphics Workstations
ISO	International Standardization Organization
IWF	Institut für Werkzeugmaschinen und Fabrikbetrieb der Technischen Universität Berlin
Linux	ein UNIX Betriebssystem auf Personal Computer
Motif	Entwicklungssystem für Benutzungsoberflächen auf X-Windows-Basis der Fa. OSF
MS-Windows	graphisches Einbenutzer-Fenstersystem der Microsoft Corp. basierend auf MS-DOS
NC	Numerical Control
NPU	Northwestern Polytechnical University in China

NT	New Technology
NURBS	Non-Uniform Rational B-Spline
RID	Rechnerinterne Darstellung
Stixel	ein zusammengesetztes Kunstwort aus den Begriffen ‚stick‘ und ‚element‘
TUB	Technische Universität Berlin
UNIX	portables Mehrbenutzer-Betriebssystem
Voxel	ein zusammengesetztes Kunstwort aus den Begriffen ‚volumen‘ und ‚element‘
VCM	Virtual Clay Modelling System
Windows 95	Weiterentwicklung von MS-Windows
Windows NT	netzwerkfähiges 32-Bit-Multitasking-Betriebssystem der Microsoft Corp.
X (Window)	portables graphisches Fenstersystem des MIT
Z-Buffer	Erweiterter Frame Buffer
G-Buffer	Geometrischer Buffer

1 Einleitung und Zielsetzung

Zur Steigerung der Produktivität und Produktqualität sowie Flexibilität setzt die Industrie in zunehmendem Maß numerisch gesteuerte Fräsmaschinen ein. Eine entscheidende Voraussetzung ist die rechtzeitige Bereitstellung fehlerfreier NC-Programme. Da eine absolute Fehlerfreiheit in den Schritten Konstruktion und NC-Planung insbesondere bei komplexen Bauteilen nicht gewährleistet werden kann [1, 2], besteht ein Bedarf insbesondere für drei- und fünfachsiges Fräsen vorab die Prozessqualität sicherzustellen. Um eine Sicherheit gegen fehlerhafte Programme zu haben, muss der Ablauf der erzeugten Programme überprüft werden. Die NC-Simulation bietet sich als Mittel zur Verifikation von Bearbeitungsabläufen, beispielsweise von NC-Werkzeugmaschinen oder Robotern an [3].

Unter NC-Simulation versteht man die rechnerunterstützte Nachbildung von NC-gesteuerten mehrachsigen Bearbeitungsprozessen zur Gewinnung von Erkenntnissen, die auf die Wirklichkeit übertragbar sind. Die NC-Simulation realer Abläufe auf Rechenanlagen setzt die Aktualisierung der Werkstückgeometrie entsprechend den Bearbeitungsanweisungen voraus. Als Lösungsansätze zur Erfüllung dieser Anforderung werden zwei Simulationsmethoden eingesetzt: nämlich direkte Volumenmodellierung und räumliche partitionierte Repräsentation. Die erste Methode basiert hauptsächlich auf Constructive Solid Geometry (CSG) und Boundary Representation (Brep). Die simulierte Materialabtragung bei der NC-Fräsbearbeitung ist durch eine Reihe von regulären Booleschen Subtraktionen zu verwirklichen. Diese Methode kann theoretisch die NC-Simulation im Solidmodellierungssystem exakt durchführen. Aber die Anwendung dieser Methode lässt sich bei der Modellierung komplexer Zerspanungsvolumen in der multiachsigen NC-Simulation und auch bei deren Booleschen Operationen beschränken [4]. Diese Hauptnachteile der ersten Methode motiviert die Anwendung der räumlichen partitionierten Methoden zur NC-Simulation.

Die räumliche partitionierte Repräsentation zur Darstellung des simulierten Werkstücks von NC-Fräsbearbeitung diskretisiert den 3D-Raum des Werkstücks in die fundamentalen Elemente, welche eine einfache Datenstruktur besitzen, zum Beispiel Z-Buffer- [5], G-Buffer- [6], Dixel- [7, 8, 9], Octree-Datenstruktur [10, 11] und Ray-Representation [12, 13, 14]. Diese Methoden ermöglichen eine einfache geometrische Boolesche Operation zwischen dem Werkstück und komplizierten Zerspanungsvolumen von NC-Simulation. Die Berechnungen der simulierten Werkstückaktualisierung werden entweder im Bildraum des Werkstücks oder geometrisch im Objektraum durchgeführt. Die Ermittlung in einem erweiterten Z-Buffer der Arbeitsstation ist eine rein graphische Lösung [5]. Ein direkter Nachteil besteht darin, dass ein Wechsel des Betrachterstandortes eine Neuberechnung der Visualisierungsdaten erforderlich macht. Geometrische Verfahren partitionieren das Werkstück

in volumetrische oder vektorische Grundelemente wie Strahlen [15, 16], Octree und Dixel, auf denen dann Boolesche Operationen ausgeführt werden. In den Strahlen-Verfahren und Dixel-Verfahren soll die Richtung der Strahlen nach der Oberflächenform des Werkstücks festgelegt werden. Die beiden Methoden sind für komplizierte NC-Fräsbearbeitung (Multi-achsiges NC-Fräsen) nicht geeignet [17].

Zahlreiche NC-Simulationssysteme sind bisher entwickelt worden. Einige von ihnen sind schon seit mehreren Jahren im Einsatz [18]. Beispielsweise bieten derzeit verfügbare NC-Simulationssysteme wie Vericut [19], dCADE [20], Hypermill [21] und WorkNC [22] umfangreiche Methoden für drei- und fünfachsig Simulation von Fräsbearbeitungsvorgängen an. Diese Systeme haben eine Gemeinsamkeit. Die NC-Simulation wird auf Basis von Konturlinien oder Dreiecksnetzen und nicht auf einem geometrischen Volumenmodell durchgeführt. Aufgrund der internen Repräsentation kann in dieser Methode nur eine 2.5D-Visualisierung realisiert werden. Ein direkter Nachteil ist, dass der Wechsel des Betrachterstandortes eine Neuberechnung der Visualisierungsdaten erforderlich macht.

Die Simulation realer Abläufe der multiachsigen NC-Fräsbearbeitung auf Rechenanlagen stellt hohe Anforderungen an die Verfahren der NC-Simulation und die Leistungen der Rechenanlagen [23]. Neue innovative Verfahren der rechnerunterstützten 3D-Modellierung und die verstärkten Leistungen der Rechner ermöglichen eine realitätsnahe NC-Simulation des Materialabtragungsvorgangs des Werkstücks auf Rechenanlagen. Der am IPK Berlin entwickelte 3D-Kernmodellierer des Virtual Clay Modelling Systems bietet den Konstrukteuren die Möglichkeit, dass das beliebige geometrische Modell im 3D-Raum mit Voxelmodell modelliert, Boolesch manipuliert und graphisch visualisiert werden kann [24]. Dieser ermöglicht die Modellierung und die Aktualisierung von komplizierten Werkstücken und Zerspanungsvolumen in der multiachsigen NC-Fräsbearbeitung auf Rechenanlagen. Aufgrund der inneren Datenstruktur des Voxelmodells sowie der hohen Leistungen der Rechenanlagen wird erreicht, dass die Nachbildung der multiachsigen NC-Fräsbearbeitung auf Rechenanlagen realitätsnah erfolgt.

Ziel der vorliegenden Arbeit ist es, eine voxelbasierte Methode für drei- und fünfachsig Simulation des NC-Fräsens zu entwickeln. Der am IPK Berlin entwickelte 3D-Kernmodellierer des Virtual Clay Modelling Systems soll zur NC-Simulation eingesetzt und erweitert werden, um die Anforderung an Arbeitsspeicher des rechnerinternen Werkstücks zu verringern und die Simulationsfehler zu kontrollieren. Die Werkstückaktualisierung wird im geometrischen 3D-Raum durchgeführt. Das simulierte Werkstück kann aus unterschiedlichen Richtungen im 3D-Raum ohne weitere Berechnungen jeder Zeit betrachtet werden. Die 3D-Informationen wie das Zerspanungsvolumen einschließlich aller Zwischenergebnisse sind jederzeit bekannt, und unterstützen durchgehend die Anwender bei

der Simulation. Die Materialverletzungen und verbliebene Restmaterialien auf dem simulierten Werkstück können im 3D-Raum zur Bewertung der Qualität der NC-Programmierung ermittelt werden. Eine effektive Optimierung der NC-Programmierung in der Fertigungsvorbereitung soll ebenso möglich sein.

Die entwickelte Methode wird zur Simulation der fünfachsigem NC-Fräsbearbeitung eines Verdichtterlaufgrades eingesetzt, um die 3D-Informationen zur Überprüfung und Optimierung von NC-Programmen zu gewinnen. Damit sollen die Anwender in die Lage versetzt werden, eine vollständige Nachbildung von NC-Fräsen im 3D-Raum vor der NC-Bearbeitung zu simulieren und danach ein fehlerfreies NC-Fräsen durchführen zu können.

2 Stand der Technik

2.1 Allgemeine Betrachtung

In diesem Kapitel sollen zu Beginn die Grundlagen zum Verständnis der vorliegenden Arbeit gelegt werden. Das betrifft zuerst die Methoden zur Modellierung von Werkstücken. Zum zweiten sollen die Methoden zur Visualisierung von Werkstücken im Vordergrund stehen.

Im weiteren Verlauf des Kapitels wird auf die vorhandenen NC-Simulationssysteme eingegangen. Zum Schluss werden die Anforderungen an die Methode zur fünfachsigen Simulation des NC-FräSENS gestellt.

Zur Fertigung geometrisch komplizierter Werkstücke werden heute zunehmend NC-Maschinen mit drei bis fünf simultan steuerbaren Achsen eingesetzt. Eine entscheidende Voraussetzung ist die rechtzeitige Bereitstellung fehlerfreier NC-Programme. Ein NC-Programm kann entweder aus einem Teileprogramm, wie APT und EXAPT oder mit Hilfe moderner CAD-Systeme wie UNIGRAPHICS, PRO-ENGINEER, CATIA, WORK-NC usw. erstellt werden. In beiden Fällen können Programmierfehler trotz sorgfältiger Überprüfung nicht ausgeschlossen werden. Die Programme sind besonders bei fünfachsiger NC-Bearbeitung schwer zu überprüfen, da es hier für den Programmierer fast unmöglich ist, sich vorzustellen, wie die in CLDATA definierte Werkzeugbahn zu den entsprechenden Achsenbewegungen des NC-Werkzeugs zugeordnet ist. Daher ist es wünschenswert, vor der Belegung von einer Maschine die realen Abläufe der NC-Bearbeitung durch ein dreidimensionales grafikunterstütztes Simulationssystem zu prüfen.

Unter NC-Simulation versteht man die rechnerunterstützte Nachbildung von NC-gesteuerten mehrachsigen Bearbeitungsprozessen zur Gewinnung von Erkenntnissen, die auf die Wirklichkeit übertragbar sind. Die NC-Simulation realer Abläufe auf Rechenanlagen setzt die Aktualisierung der Werkstückgeometrie entsprechend den Bearbeitungsanweisungen voraus. Für eine komplexe simulierte Werkstückaktualisierung spielt die rechnerinterne Gestaltung des Zerspanungsvolumens des Werkzeugs gemäß dem fünfachsigen NC-Programm eine große Rolle. Das Zerspanungsvolumen ist der räumliche Bereich, den das Werkzeug in einem Bewegungsintervall (entsprechend einem NC-Satz) im 3D-Raum durchfahren hat [2]. Im Verlauf der fünfachsigen NC-Bearbeitung erzeugt ein bewegtes Werkzeug im 3D-Raum ein komplex geformtes Zerspanungsvolumen. Die Bearbeitung des Werkstücks kann durch die Subtraktion des Zerspanungsvolumens des Werkzeugs aus der Geometrie des Werkstücks rechnerisch nachgebildet werden. Die grafische und geometrische Abbildung des Bearbeitungsvorganges und der Zwischenzustände des Werkstücks erlauben die Verifikation kom-

plexer Werkzeugwege, der Bearbeitungsergebnisse und die Kollisionskontrolle zwischen dem Werkstück und dem Werkzeug. Die wichtigsten Aufgaben der simulierten Werkstückaktualisierung sind:

- Rechnerinterne Modellierung des simulierten Werkstücks,
- Rechnerinterne Gestaltung von Zerspanungsvolumen,
- Boolesche Subtraktion zwischen Werkstück und Zerspanungsvolumen,
- Ermittlung der Kollisionsbereiche,
- Visualisierung des simulierten Werkstücks.

Bild 2.1 zeigt die simulierte Werkstückaktualisierung.

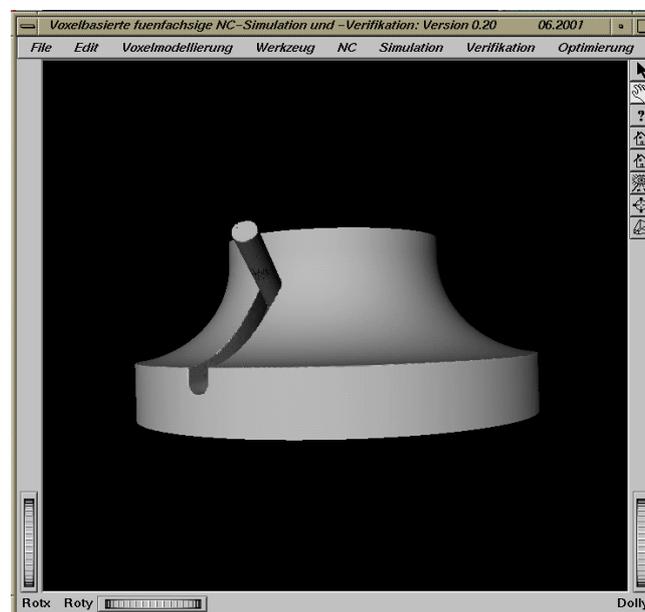


Bild 2.1: Fünfachsigige NC-Simulation eines Verdichterlaufrades

2.2 Volumenorientierte Repräsentationen

2.2.1 Direkte Volumenrepräsentationen

Für die simulierte Werkstückaktualisierung werden die volumenorientierten Repräsentationen in den derzeit angewendeten NC-Simulationssystemen eingesetzt. Bei volumenorientierten Repräsentationen kann wieder zwischen direkten Volumenrepräsentationen und räumlichen partitionierten Repräsentationen unterschieden werden [7].

Die direkten Volumenmodellierungsmethoden werden in akkumulative, generative und hybride Methoden klassifiziert. Bei der akkumulativen Methode setzt sich das Objekt aus einer Summe von Grundelementen zusammen [25, 26, 27, 28]. Beim wichtigsten Vertreter, der B-Rep - Boundary Representation, bestehen diese Elemente aus Grenzflächen vom umgebenden Raum. Bei der Datenstruktur wird eine Trennung in geometrische und topologische Daten vorgenommen [29]. Die geometrischen Informationen über Grenzflächen werden mit Hilfe der analytischen Geometrie dargestellt. Durch die topologischen Daten werden die Definitionsbereiche eines Objekts festgelegt. Eingabe und Manipulation der Modellinhalte erfolgen häufig über Mengenoperationen, sogenannte Boolesche Operationen auf Basisvolumen, die intern algorithmisch in die Modellierprimitiva zerlegt, verknüpft und in der Datenstruktur ausgewertet, abgelegt werden. Die Volumen liegen in der Datenstruktur in evaluierter, expliziter Form vor, die einzelnen Grundelemente – Flächen, Kanten und Punkte – können daher direkt ausgewertet und manipuliert werden. Eine vollständige Integration frei geformter Grundelemente ist, unabhängig von ihrem Typ, bislang jedoch nicht gelungen. Gründe hierfür sind Genauigkeitsprobleme frei geformter Übergänge und die regelmäßig auftretenden Sonderfälle der Schnittberechnungen [30].

Die generative Methode, wie CSG - Constructive Solid Geometry, bildet Körper mit der Speicherung von Verknüpfungsoperationen und Grundkörpern als Operanden ab und ist daher generierend zum Zeitpunkt eines Modellzugriffs [24]. Die bekanntesten Grundkörper sind Volumenprimitive wie Quader, Zylinder, Kegel, Kugel und Torus. Als Verknüpfungsoperationen werden normalerweise die Booleschen Operationen Addition, Subtraktion und Durchschnitt sowie Transformationen, bei denen Lage und Orientierung verändert werden, betrachtet [31]. Komplexe Körper liegen in der Datenstruktur in nicht evaluierter, impliziter Form als eine Sequenz Boolescher Operationen auf Grundkörper vor, die in der Datenstruktur als algebraische Ausdrücke in binären Bäumen abgelegt werden können. Änderungen an einem Objekt sind sehr rechenzeitintensiv, da für jede Änderung die binäre Baumstruktur des Objekts neu ausgewertet wird [3]. Zum Zeitpunkt des Modellzugriffs ist auch ein höherer Berechnungsaufwand zu leisten, da Geometriebeschreibungen in der Datenstruktur als algebraische Ausdrücke in binären Bäumen abgelegt sind und das Modell noch generiert werden muss.

Unter einem hybriden Modell soll ein Modell verstanden werden, das Komponenten verschiedener Modelltypen enthält. Das wichtigste hybride Modell ist die in heutigen CAD-Systemen eingesetzte Kombination eines akkumulativen (B-Rep) und eines generativen Modells (CSG). Jedoch ist mit einem hybriden Ansatz nicht zwangsläufig die Summe der günstigen Eigenschaften der einzelnen Repräsentationsformen zu erreichen. Im Gegenteil, ungünstige Eigenschaften kumulieren in hybri-

den Ansätzen, wenn ein vergleichbarer mathematischer Modellraum realisiert werden soll [24]. Darüber hinaus ist für eine wechselseitige Modellierung auf der einen wie auch der anderen rechnerinternen Darstellung zur Wahrung der Konsistenz jedwede Änderung der rechnerinternen Darstellung in die jeweils andere umzusetzen. Um diesen Umstand zu beseitigen, sollten nur einzelne rechnerinterne Datenstrukturen wie B-Rep- oder CSG-Datenstruktur benutzt werden.

2.2.2 Räumliche partitionierte Repräsentationen

2.2.2.1 Explizite Zellrepräsentation

Die räumliche partitionierte Repräsentation zur Darstellung des simulierten Werkstücks von NC-Fräsbearbeitung diskretisiert den 3D-Raum des Werkstücks in die fundamentalen grafischen Elemente, welche einfache Datenstrukturen besitzen, zum Beispiel Zelldarstellung, Z-Buffer-, G-Buffer, Dixel-, Octree-Datenstruktur und Ray-Repräsentation. Im Folgenden werden vier verschiedene Ansätze der räumlichen partitionierten Repräsentationen vorgestellt, nämlich die explizite Zellrepräsentation, das Octree-Verfahren, Voxel mit Residuen-Verfahren und Dixel-Verfahren.

Bei Zelldarstellung wird der 3D-Raum, in dem sich das zu repräsentierende Objekt befindet, in identische, innerhalb eines gleichmäßigen Gitters angeordneter Zellen aufgeteilt [31]. In üblicher Weise lassen sich die Gitter im 3D-Raum mittels drei Strahlen, die jeweils zu entsprechender Koordinatenachse parallelisiert werden, gliedern (Bild 2.2). Ein volumetrisches Objekt wird durch diskrete Bausteine, beispielsweise Würfel einheitlicher Größe, in binären Zuständen als Teil oder außerhalb des Körpers approximiert. Die Genauigkeit des abgebildeten Objekts hängt vom Diskretisierungsgrad ab. Mit der Diskretisierung gehen geometrische Zusammenhänge verloren, da jede Zelle im einfachsten Fall nicht mit mehr als ihrer Lage abgebildet wird. Die verbliebenen topologischen Informationen über innere benachbarte Körperpunkte sind in der Datenstruktur (i, j, k, m) festgehalten. Der Parameter m gibt an, ob eine bestimmte Zelle (i, j, k) im dreidimensionalen Raum innerhalb ($m=1$) oder außerhalb eines Körpers liegt. Daher wird eine selektive Aktualisierung einzelner Zellen ermöglicht, ohne dass die Zellenstruktur grundlegend geändert wird. Eine weitgehende Flexibilität der Körpergestaltung ist daher möglich. Bei Zellmodellen gibt es folgende Probleme:

- Raumorientierte Zellmodelle basieren auf einer Diskretisierung geometrischer Formen. Sie sind daher approximierend mit einer Genauigkeit im Bereich der kleinsten Teilungstiefe,
- Oberflächeninformationen und damit Normalen sind nicht mehr verfügbar,

- In einem expliziten Zellmodell mit einer in alle drei Raumrichtungen gleichen Ausdehnung wächst die Anzahl der Zellen und somit der Speicherplatzbedarf kubisch im Verhältnis zur Anzahl der entlang einer Kante des Zellraums befindlichen Zellen. In der expliziten Form sind Zellmodelle sehr speicherplatzintensiv.

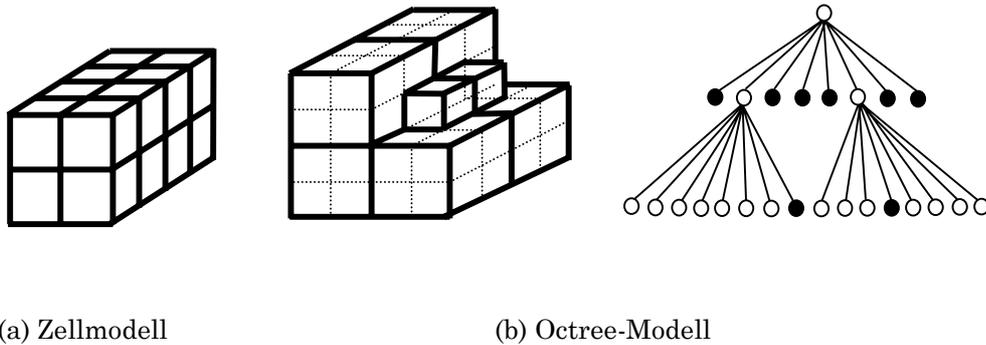


Bild 2.2: Zellmodell und Octree-Modell

2.2.2.2 Das Octree-Verfahren

Beim Octree-Verfahren wird zuerst der gesamte Voxelraum in acht kleinere Subwürfel zerlegt. Diese Subwürfel, die als voll (oder leer) klassifiziert werden, also ganz innerhalb (oder ganz außerhalb) des Objekts liegen, werden nicht weiter unterteilt. Nur dort, wo teilweise ausgefüllte Subwürfel liegen, werden diese weiter unterteilt. Dies Prozess wird solange fortgesetzt, bis eine zufriedenstellende Auflösung erreicht ist.

Der Vorteil des Octree-Verfahrens liegt darin, dass große, homogene Bereiche durch einen einzigen Subwürfel repräsentiert werden, so dass im Vergleich zum Zellmodell Speicherplatz gespart wird. Auch lässt sich das Octree-Verfahren vorteilhaft für manche Visualisierungsalgorithmen nutzen, da sich große, homogene Räume, die durch ein einziges Element repräsentiert werden, schnell verarbeiten lassen.

Im Octree-Verfahren wird die Teilungsbedingung für einen Subwürfel zu einer Homogenitätsbedingung. Ein Subwürfel wird dann nicht weiter zerlegt, wenn die vorgegebene minimale Auflösung erreicht ist. Natürlich führt das – abhängig von der Verteilung von Materialdaten im Voxelraum – zu einer immer größer werdenden Anzahl von Subwürfeln. Von einem bestimmten Punkt an ist daher die Repräsentation durch Octree-Verfahren bezüglich des Speicherbedarfs unterlegen.

Nachteilig ist die Abhängigkeit der Struktur eines Octrees von der Lage des zu repräsentierenden Objekts im 3D-Raum. Wird ein Objekt transformiert, so führt das zu teilweise nicht mehr nachvoll-

ziehbaren Änderungen des Octrees. Daher lässt sich das Octree-Verfahren eher für statische Repräsentationen, wie es beispielsweise für die Visualisierung notwendig ist, sinnvoll verwenden [32]. Dieser Nachteil macht sich auch besonders bei kleinteiligen Änderungen der Voxeldaten bemerkbar. Sollen einzelne Voxel verändert werden, ist es unter Umständen nötig, den gesamten Bereich, in dem die Voxel liegen, neu zu segmentieren.

Diese Trägheit gegenüber kleinteiligen Änderungen in den Voxeldaten macht sich besonders bei simulierter Werkstückaktualisierung, die bei der dynamischen Änderbarkeit des Geometriemodells im Vordergrund steht, bemerkbar.

2.2.2.3 Voxel mit Residuen

Eine weitere Möglichkeit zur Begrenzung des Speicherplatzbedarfs einer Voxeldarstellung, ohne auf eine hohe Genauigkeit in der Darstellung verzichten zu müssen, besteht darin, den Voxeln Zusatzinformationen für eine verbesserte Darstellung hinzuzufügen. Zum Beispiel können den Randvoxeln eines Volumenkörpers im Voxelraum zusätzliche Residuen hinzugefügt werden. Diese Residuen sind Informationen über die Approximationsungenauigkeit der relativ grob aufgelösten Voxeldarstellung gegenüber einem genaueren Oberflächenmodell (Bild 2.3 a). Jedem Voxel können bis zu sechs unterschiedliche Residuenwerte hinzugefügt werden, von denen jeder den Abstand des Mittelpunkts des durch den Voxel repräsentierten Würfels zu der tatsächlichen Oberfläche des dargestellten Volumens entlang einer der drei Raumachsen in positiver oder negativer Richtung darstellt. Ein bestehendes Volumenmodell wird also durch zusätzliche oberflächenorientierte Parameter partiell höher aufgelöst. Bereits bei einer geringen Diskretisierung des Zellmodells ist eine gute Approximation erreichbar. Im Modell des Voxels mit Residuen fasst die Lauflängenkodierung Zellen des gleichen Informationsgehalts in einer Koordinatenrichtung zusammen. Im Vergleich zur Octree-Repräsentation konnte gezeigt werden, dass mit der Lauflängenkodierung eine höhere Datenkomprimierung erreicht werden kann [33]. Ein weiterer Vorzug der Lauflängenkodierung ist die mögliche Zusammenfassung des Vergleichs von Zelle zu Zelle entsprechend der Lauflänge und daher die mögliche Beschleunigung von Basisalgorithmen, die zur Verzeichnung von Modelländerungen angewendet werden [24]. Bei Voxel mit Residuen-Verfahren gibt es folgende Probleme:

- Hoher Speicherplatzbedarf, da neben Lauflängeninformatoren die Residuen in der Datenstruktur gespeichert werden müssen,
- Die Oberflächenpunkte eines Körpers sind indirekt durch Lauflängeninformatoren und entsprechende Residuen beschrieben. Zur Aktualisierung eines zu verändernden Körpers müssen dessen Oberflächeninformationen erzeugt werden.

Das Voxel mit Residuen-Verfahren wurden in dem am IPK Berlin entwickelten 3D-Kernmodellierer des Virtual Clay Modelling Systems eingesetzt [24]. Das Virtual Clay Modelling System wird später vorgestellt.

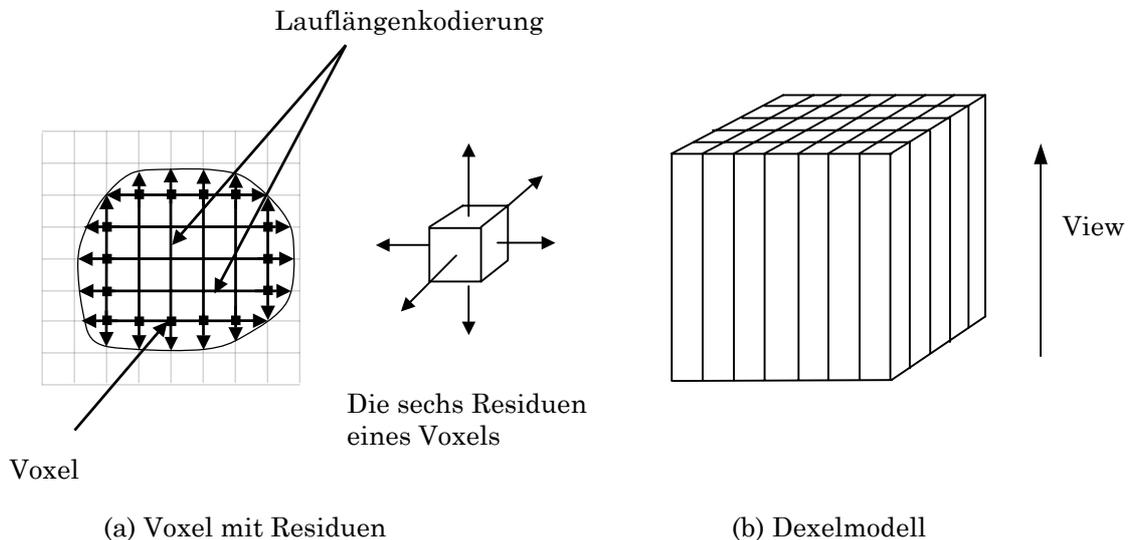


Bild 2.3: Voxel mit Residuen und Dexelmodell

2.2.2.4 Dexel¹ – Repräsentation

Ausgehend von dem Prinzip des Z-Buffers hat Van Hook zuerst den Begriff Dexel für NC-Simulationen eingeführt [5]. Weiter wurde die Dexel-Repräsentation von Hui [8], Huang und Olive [7], und Stifter [34, 35, 36] bei drei- und fünfsachsiger NC-Simulation verwendet.

Bei der Dexel-Repräsentation werden nicht mehr die einzelnen Voxel gespeichert, sondern nur noch die Positionen der Materialwechsel. Dazu werden von einer Ebene (wie XY-Ebene) senkrechte Strahlen in die Szene gelegt. Diese Strahlen haben zwangsläufig einen diskreten Abstand zueinander. Aber entlang der Strahlen können die Oberflächenpositionen mit Fließkommagenauigkeit gespeichert werden. In der Dexel-Repräsentation wird völlig darauf verzichtet, Voxel oder Residuen zu ermitteln (Bild 2.3 b). Im Vergleich zur Octree-Repräsentation und Voxel mit Residuen-Repräsentation konnte gezeigt werden, dass mit der Dexel-Repräsentation eine höhere Datenkomprimierung erreicht werden kann [24]. Da eine Kodierung auf der Ebene der Dateneinträge erfolgt, hat die Dexel-Repräsentation eine flache Struktur, die eine selektive Aktualisierung des einzelnen Dexels ermöglicht, ohne dass die Dexelstruktur grundlegend verändert wird.

¹ Dexel ist ein aus den Begriffen ‚depth‘ und ‚element‘ zusammengesetztes Kunstwort.

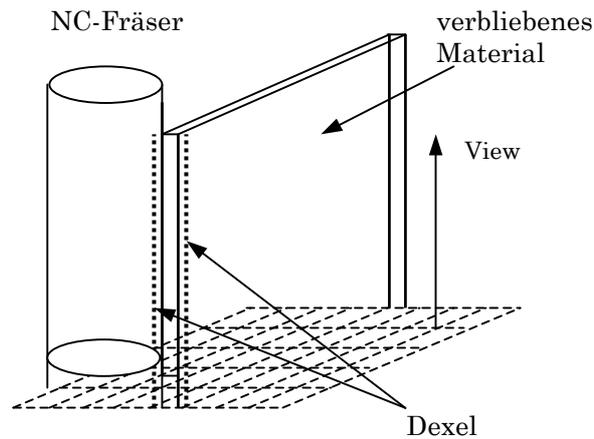


Bild 2.4: Beispiel zur Darstellung des Problems bei der dexelbasierten NC-Simulation

Für die NC-Simulation hat die Dixel-Repräsentation folgende Probleme:

- Die Genauigkeit des mittels Dixel-Modell repräsentierten Werkstücks hängt direkt von der Richtung und Größe des Dexels ab. Somit muss vom Anwender zuerst die Dixel-Richtung ausgewählt werden, um die bessere Modellierungsgenauigkeit des Werkstücks zu erreichen [7]. Die Richtung des Dexels ist bei komplex geformtem Werkstück, wie bei fünfschiger NC-Simulation, unter Umständen schwer zu finden.
- Zugunsten der Visualisierung des simulierten Werkstücks sind Ray-Richtung und Dixel-Richtung identisch [5, 7, 8, 34]. Der Wechsel des Betrachterstandortes macht eine rechnerinterne Neumodellierung des simulierten Werkstücks und eine Neuberechnung der Visualisierungsdaten erforderlich.
- In bestimmten Situationen versagt die dexelbasierte NC-Simulation. Bild 2.4 zeigt ein solches Beispiel. Die NC-Simulation hat gezeigt, dass das Material auf dem simulierten Werkstück vollständig gefräst werden sollte, während das Scheibenmaterial auf realem Werkstück erhalten blieb. Stifter [34] hat eine Methode zur Beseitigung dieses Problems entwickelt. Mehr als zwei View-Richtungen wurden ausgewählt und entsprechend der einzelnen View-Richtung wurden mehrere rechnerinterne Dixelmodelle des simulierten Werkstücks gestaltet. Dieses Problem kann mit dem in der Arbeit verwendeten Stixelmodell überwunden werden.

2.2.3 Zusammenfassung

Nach dem Stand der Forschung und Technik sind keine Verfahren bekannt, die zeitlich veränderte simulierte Werkstückzwischenzustände als räumliche partitionierte Volumenmodelle im Objekt-raum sowohl für die dreiachsige wie auch die fünfachsiges Fräsbearbeitung liefern. Die Dixel-Methode gehört zu den räumlichen partitionierten Volumenmethoden und besitzt hervorragende Eigenschaften wie geringer Speicherplatzbedarf bei bestimmter Auflösung und keine echten Booleschen Operationen bei der Materialabtragung. Somit wurde die Dixel-Methode in den heutigen drei- und fünfachsiges NC-Simulationssystemen eingesetzt [8, 34, 37]. Trotzdem sind diese NC-Simulationssysteme durch die rechnerinterne Dixel-Darstellung beschränkt. In der vorliegenden Arbeit wurde eine neue voxelbasierte Methode zur Repräsentation des simulierten Werkstücks für die fünfachsiges NC-Simulation verwendet.

2.3 Zerspanungsvolumen

Die Gestaltung des Zerspanungsvolumens des Werkzeugs entlang der NC-Bahnen ist ein wichtiger Schritt in der fünfachsiges NC-Simulation. Unter einem werkzeugbezogenes Zerspanungsvolumen ist der räumliche Bereich zu verstehen, den das Werkzeug innerhalb eines Bewegungsintervalls (entsprechend einem NC-Satz) im Raum überstreicht. In Bild 2.5 ist ein werkzeugbezogenes Zerspanungsvolumen dargestellt.

Bei der Bewegung des Werkzeugs verändern sich die Position des Werkzeugs und die Richtung der Werkzeugachse. Diese Veränderung kann durch eine Funktion, die Bewegungsfunktion genannt wird, dargestellt werden. Das Zerspanungsvolumen lässt sich durch die Geometrie des Werkzeugs und die Bewegungsfunktion festlegen [38]. Jedoch für die verschiedenen Werkzeugsformen und Bewegungsfunktionen ist keine einheitliche implizite, geschlossene Darstellungsform für das Zerspanungsvolumen vorhanden. Eine einfache Lösung für diese mathematische Schwierigkeit ist die direkte Durchführung der Booleschen Subtraktion zwischen dem bewegenden Werkzeug und dem Werkstückmaterialmodell durch intensive Diskretisierung des Zeitraums des Bewegungsintervalls [8, 39]. Ein solches System wurde bei General Dynamics zur Verifikation der NC-Bahnen entwickelt. Die Nachteile dieser Methode sind die Zeitaufwendigkeit und die Anwendung eines CAD-Volumenmodellierungssystems.

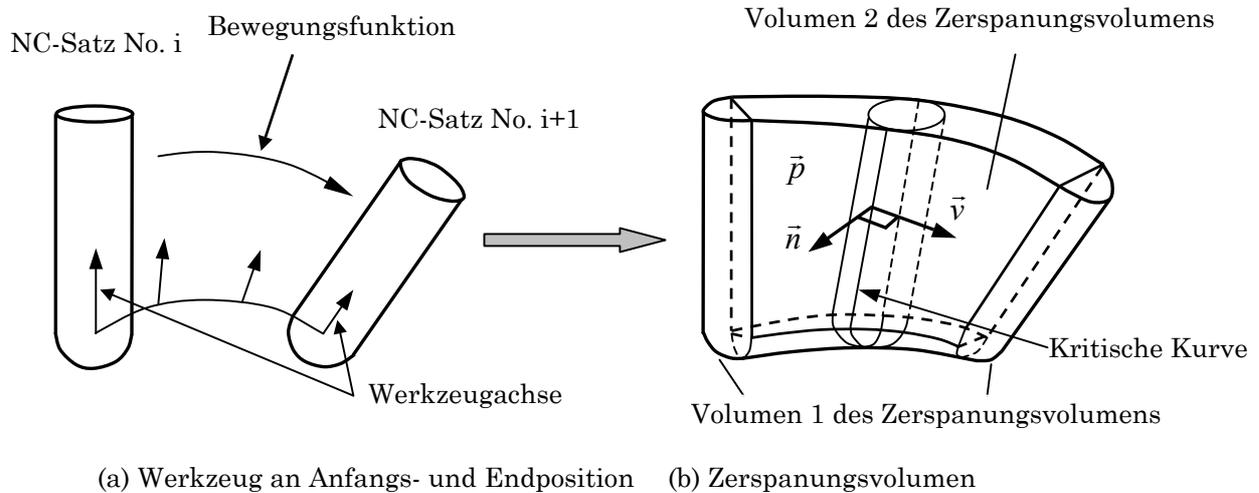


Bild 2.5: Die Bewegungsfunktion eines fräsenden Werkzeugs

Zur Vermeidung dieser Schwierigkeiten werden die flächenbasierten Modellierungsmethoden für Zerspanungsvolumen entwickelt. Statt der Modellierung des Zerspanungsvolumens wird die Grenzfläche des Zerspanungsvolumens, die als Swept-Envelope bezeichnet wird, gestaltet. Die Swept-Envelope des Werkzeugs innerhalb eines Bewegungsintervalls wird in zwei Kategorien eingeteilt. Die eine entspricht den Teilflächen des Werkzeugs in Anfangs- und Endposition des Bewegungsintervalls, die andere den neuen von den Außenkanten des Werkzeugs innerhalb des Bewegungsintervalls erzeugten Flächen. Da die erste Kategorie durch vorhandene Modellierungsmethoden dargestellt werden kann, haben sich viele Forschungstätigkeiten zur Modellierung des Zerspanungsvolumens üblicherweise nur auf die zweite Kategorie konzentriert.

Ausgehend von der Envelope-Theorie haben Wang und Wang [40, 41, 42] eine allgemeine Methode zur Gestaltung der Envelope-Flächen des Zerspanungsvolumens des Werkzeugs entwickelt. Durch Beschreibung des Werkzeugs und der Bewegungsfunktion werden die Envelope-Flächen implizit dargestellt. Diese Methode geht davon aus, dass das Werkzeug die Envelope-Flächen des Zerspanungsvolumens entlang einer kritischen Kurve (siehe Bild 2.5) auf der Oberfläche des Werkzeugs tangiert. Die kritischen Kurven lassen sich unter der Bedingung

$$\vec{n}_{(\vec{p})} \bullet \vec{v}_{(\vec{p})} = 0 \quad (2-1)$$

ermitteln, wobei $\vec{n}_{(\vec{p})}$ die Oberflächennormale des Werkzeugs an Punkt \vec{p} und $\vec{v}_{(\vec{p})}$ die Geschwindigkeit des Oberflächenpunkts darstellt. Diese kritischen Kurven bilden somit das Außenprofil des Werkzeugs, das von der Bewegungsrichtung des Werkzeugs aus betrachtet wird. Um ein rechnerin-

tern gültiges Modell für das Zerspanungsvolumen zu erstellen, bleiben die kritischen Kurven innerhalb des Bewegungsintervalls topologisch unverändert. Zur Ermittlung eines Punkts auf der Envelope-Fläche des Zerspanungsvolumens ist jedoch normalerweise eine nichtlineare Formel zu lösen [38], hier ist üblicherweise eine numerische Technik erforderlich. Bild 2.5 zeigt ein Beispiel eines solchen Zerspanungsvolumens eines Werkzeugs.

Ein anderer Ansatz zur Modellierung des Zerspanungsvolumens des Werkzeugs, der als Reduktionsmethode bezeichnet wird, wurde von Ganter & Uicker [43] and Leu, Park, Wang [44] 1986 zuerst vorgestellt. Ganter und Uicker definierten die bewegte Silhouette des Zerspanungsvolumens eines Werkzeugs im 3D-Raum als die Teilmenge von Grenzpunkten des innerhalb des Bewegungsintervalls bewegten Werkzeugs. Zur Annäherung des Zerspanungsvolumens des Werkzeugs mit diesem Ansatz hat man einen Set von Punkten auf der Oberfläche des Zerspanungsvolumens zu berechnen. Viele Weiterentwicklungen dieses Ansatzes wurden bis vor kurzem durchgeführt [45, 46, 47].

Die Swept Differential Equation (SDE) Methode [48, 49], die sowohl zur Envelope-Theorie als auch zur Reduction Methode gehört [50], wurde zur Modellierung des Zerspanungsvolumens eines Werkzeugs entwickelt. Diese SDE-Methode basiert auf Differenzialgeometrie, Differenzialtopologie und der Theorie der Lie Groups, die von do Carmo [51], Guilleman & Pollack [52] and Warner [53] beschrieben wird. Zur Verbesserung des Zeitbedarfs dieser Algorithmen bei der Berechnung wurde die SDE-Methode in die Swept-Envelope-Differential Equation (SEDE)-Methode erweitert [54, 55]. Ein auf SEDE basiertes NC-Simulationssystem, das mit kommerzieller NC-Software gekoppelt wird, wurde von Blackmore und seinen Kollegen zur Simulation und Verifikation von NC-Programmen entwickelt.

Im Rahmen der vorliegenden Arbeit wurde die allgemeine Envelope-Flächen-Methode, wie die Wang-Wang-Methode [40], zur Gestaltung von Zerspanungsvolumen in der voxelbasierten NC-Simulation weiterentwickelt. Die impliziten, geschlossenen Darstellungsformeln zur Ermittlung des Zerspanungsvolumens von Kugelfräser, Zylinderfräser, Kegelfräser und Fassfräser wurden abgeleitet. Somit lässt sich die Ermittlung der Envelope-Flächen des Zerspanungsvolumens effizient durchführen. Die ermittelten, durch Oberflächen dargestellten Zerspanungsvolumen wurden wieder durch ein Voxelmodell rechnerintern nachgebildet. Es wurde nachgewiesen, dass die rechnerintern nach-

gebildeten Voxelvolumen für die Aktualisierung und Visualisierung des Werkstücks sehr praktisch sind. (siehe Kapitel 4).

2.4 Fehlerbetrachtung in der NC-Simulation

Bei der diskretisierten NC-Simulation ist das zu simulierende Werkstück mit Voxeln und dessen Flächen mit Sample-Punkten im 3D-Raum repräsentiert. Zwischen den Sample-Punkten ist keine Information über diese Fläche vorhanden. Es besteht die Möglichkeit, dass das simulierte Werkstück in jedem Sample-Punkt die vorgegebene Toleranz nicht überschreitet. Aber in anderen Orten zwischen den Sample-Punkten wird diese Fläche möglicherweise verletzt. Es ist auch damit zu rechnen, dass auf der Fläche überflüssige Materialien hinterlassen werden. Um den Simulationsergebnissen gerecht zu werden, werden die in der NC-Simulation eingeführten sogenannten Simulationsfehler in Betracht gezogen.

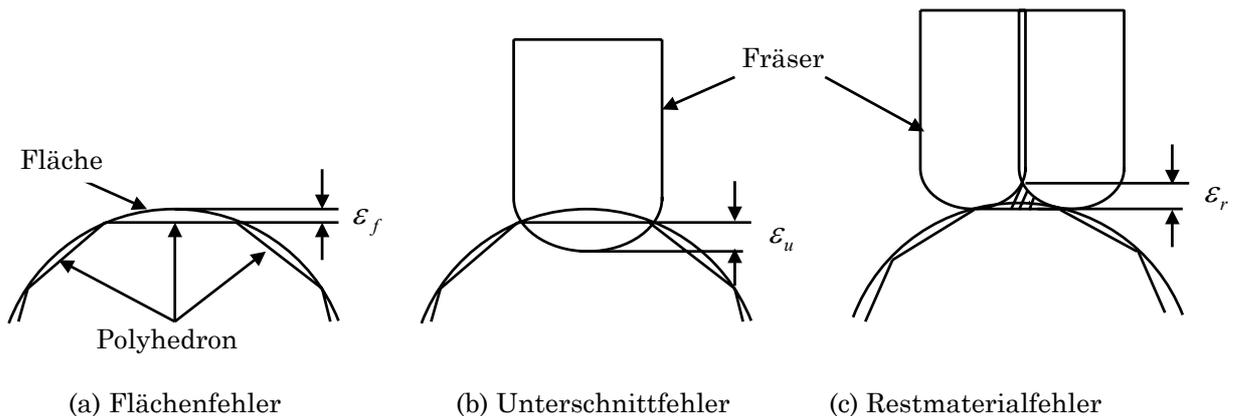


Bild 2.6: NC-Simulationsfehler

Unter einem NC-Simulationsfehler versteht man die Abweichungen, die sich zwischen der gefrästen Fläche und der Sollfläche ergeben, wenn keine Differenz zwischen diesen zwei Flächen bei den Sample-Punkten festgelegt wird [56]. Die Ursache des NC-Simulationsfehlers wird in drei Bereiche gliedert. Diese werden wie folgt ermittelt. Zum ersten wird die Flächenrepräsentation herangezogen. Die Oberfläche des simulierten Werkstücks wird durch Sample-Punkte interpoliert. In der Praxis wird die Fläche üblicherweise durch Dreiecksnetze angenähert. Diese Annäherung ist nicht exakt, da Abweichungen zwischen der Sollfläche und den Dreiecksnetzen vorliegen. In der vorliegenden Arbeit wird diese Abweichung als Flächenfehler bezeichnet. Dieser Flächenfehler hängt direkt von der Anzahl der Sample-Punkte und der Krümmung der Sollfläche ab. Nimmt die Anzahl der Sample-

Punkte innerhalb eines hoch gekrümmten Bereiches zu, so verringert sich der Flächenfehler in diesem Bereich. Bild 2.6a zeigt den Flächenfehler ε_f .

Zum zweiten wird davon ausgegangen, dass die Schnittinformationen der gefrästen Werkstückoberfläche nur in den Sample-Punkten bekannt sind. Es kann passieren, dass die Oberfläche zwischen Sample-Punkten ohne die Verletzungen der Oberfläche in den Sample-Punkten unterschritten wird. Dieser Unterschritt hängt direkt mit dem Fräserdurchmesser und dem Abstand zwischen zwei Sample-Punkten zusammen und wird Unterschchnittfehler genannt. Bild 2.6b stellt den Unterschchnittfehler ε_u dar. Hierbei wird der Kugelfräser herangezogen.

Der dritte NC-Simulationsfehler wird durch die Restmaterialien auf der Ebene eines Dreieckes verursacht. Während der Fräser die Sample-Punkte exakt durchschneidet, bleiben auf der Ebene des entsprechenden Dreieckes überschüssige Materialien, deren maximale Höhe auf dem Dreieck als Restmaterialfehler ε_r bezeichnet wird. Der Restmaterialfehler befindet sich normalerweise in der Mitte des Dreieckes. Um diesen Restmaterialfehler zu ermitteln, benötigt man zusätzliche Informationen zwischen den Sample-Punkten. Bild 2.6c zeigt den Restmaterialfehler ε_r .

Drysdale und Jerard sind auf die NC-Simulationsfehler eingegangen. Jedoch wurde die Analyse der NC-Simulationsfehler auf dreiaxige Simulationen beschränkt. Die weitere Entwicklung zur Begrenzung der NC-Simulationsfehler für fünfachsiges NC-Simulationen wurde nicht veröffentlicht. Stifter [34] und Huang [7] haben nur den Flächenfehler bei der Diskretisierung, basierend auf dem Dixelmodell, analysiert. Die Unterschchnittfehler und Restmaterialfehler wurden nicht berücksichtigt. Andere zahlreiche Systeme haben die NC-Simulationsfehler nicht herangezogen [8, 2].

Im Rahmen der vorliegenden Arbeit wurde der Simulationsfehler in der fünfachsiges NC-Simulation analysiert und die Methode zur Begrenzung des Simulationsfehlers in der voxelbasierten fünfachsiges NC-Simulation entwickelt.

2.5 Visualisierung in der Werkstückaktualisierung

2.5.1 Direktes Verfahren

Für die Werkstückaktualisierung bei der NC-Simulation hat die Visualisierung des Werkstücks zentrale Bedeutung. Die Werkstückvisualisierung basiert auf der Umsetzung der rechnerinternen volumetrischen Repräsentation in ein graphisches Bild auf dem Bildschirm. Um dem Anwender die

Vorstellung der veränderlichen Zustände des Werkstücks zu vermitteln, muss das simulierte Werkstück stetig auf dem Bildschirm dargestellt werden. Neben den Operationen auf dem Volumenmodell benötigt die Visualisierung einen großen Teil der Rechnerkapazität. Bei der echtzeitnahen Werkstückaktualisierung stellt die passende Abwägung zwischen der Qualität der Bildgenerierung und der Geschwindigkeit ein Problem dar. Heutzutage existieren Rendering-Verfahren, die eine derart optimale Bildgenerierung zulassen, so dass das Bild für das ungeübte Auge kaum noch als vom Rechner erzeugt zu erkennen ist.

Die Verfahren zur Visualisierung von Volumendaten können in zwei Kategorien eingeteilt werden [57, 58]:

- Direktes Verfahren,
- Indirektes Verfahren.

Die direkten Verfahren zur Volumenvisualisierung bilden ohne eine Konvertierung der Grundelemente Voxel, Zellen oder Zellbereiche auf der Bildebene ab. Dabei wird für jedes Pixel des Bildschirms ein Strahl zum Objekt hin verfolgt, bis er auf einen Voxel-Wert des Objekts trifft. Das Bildschirmpixel bekommt denjenigen Farbwert, der der Farbe und der Beleuchtung dieser Stelle des Objekts entspricht [59, 60]. Seit der ersten Vorstellung dieses Verfahrens, im Jahre 1985, wurden zahlreiche Weiterentwicklungen durchgeführt, wodurch eine Verbesserung und Erweiterung der ursprünglichen Verfahren erzielt wurden.

Da jedoch keine weiteren Informationen über das Objekt, wie zum Beispiel Oberflächen-Normalen, zum Zeitpunkt der Farbgebung zur Verfügung stehen, lassen sich auf diese Weise nur recht einfache Objektabbildungen vornehmen. Lichtreflektion, Texturierung oder gar Lichtbrechung können mit diesem Verfahren nicht oder nur approximativ realisiert werden. Auch aufgrund des Fehlens der Oberflächennormalen wird das Objekt unstetig gezeichnet. Die Änderungen des Betrachterstandpunkts sowie die Transformation des Volumens erfordern eine Neuberechnung der Visualisierungsdaten.

Die direkten Verfahren werden häufig im medizinischen Umfeld eingesetzt. Die Gründe dafür liegen darin, dass dort die Visualisierung nur zum Transport der gewonnenen Informationen dient und ästhetische Aspekte nicht zum Tragen kommen. Für NC-Simulationen finden diese Verfahren häufig Anwendung. Mit der erweiterten Z-Buffer-Methode [61] werden die Booleschen Operationen und die Visualisierung des Werkstücks im Bildraum durchgeführt. Die Ermittlungen der Oberflächeninformationen wie die Normalen, die Restmaterialien und die Unterschnitte sind im Bildraum unmög-

lich. Zur Beobachtung der Ergebnisse der NC-Simulation in unterschiedlichen Richtungen müssen die NC-Simulationsprozesse in mehreren Bildräumen ausgeführt und dargestellt werden. Die Umsetzung der direkten Verfahren in Hardware ist einfach.

2.5.2 Indirektes Verfahren

Die indirekten Verfahren ermitteln zunächst höherwertige geometrische Grundelemente wie polygonale Flächen oder Tetraeder, die dann konventionell mit Unterstützung von Graphiksystemen auf dem Bildschirm aufgezeichnet werden. Nach der einmaligen Ermittlung einer Berandung liegt eine sekundäre polygonale Repräsentationsform vor, welche mit unterschiedlichen Schattierungs- und Darstellungsarten sowie Transformationszuständen weiterverarbeitet werden kann, ohne den Voxel datensatz erneut traversieren zu müssen. Diese Verfahren unterscheiden sich durch den Grundelementtyp, der jeweils zur Darstellung der Oberfläche eingesetzt wird. Die Opaque-Cube-Methode [62] definiert für jede Seite einer Zelle planare Vierecke, während im Dividing-Cube-Verfahren Zellen in Pixelgröße unterteilt und als orientierte Flächenpunkte gezeichnet werden [63, 64, 65]. Das Marching-Cubes-Verfahren generiert hingegen bis zu fünf Dreiecke für eine Oberflächenzelle, die schlüssig zusammenhängend mit den Polygonen der Nachbarzellen abschließen [66, 67]. Dreiecke sind die am schnellsten zu visualisierenden Graphikformen.

Für die Werkstückaktualisierung im Einsatz sind jedoch auch Bildraumverfahren, die in direkter Weise geometrische Grundelemente des Werkstücks auswerten und auf der Bildebene abbilden. In der vorliegenden Arbeit wird das Marching-Cube-Verfahren als Basis für die Visualisierung der Werkstückaktualisierung ausgewählt, da die Oberflächen auf Dreiecksbasis generiert werden, und die Lagen der Oberflächen können anhand der Funktionswerte in Zelleckpunkten flexibel ausgerichtet werden. Die detaillierte Darstellung des Marching-Cube-Verfahrens wird im Verlauf der Arbeit vorgestellt.

2.6 NC-Simulationssysteme

Zahlreiche NC-Simulationssysteme sind bisher entwickelt worden. Viele davon sind schon seit mehreren Jahren im Einsatz. Die Anwendungsbereiche reichen sowohl vom Aufbau von Bausteinen zur Konfiguration von Bearbeitungsumgebungen, wie die Gestaltung von Werkzeugmaschinen, Werkstück-Spannmitteln [68, 69, 70, 71, 72], als auch zur Gestaltung von dem zu bearbeitenden und bearbeiteten Werkstück [73], Simulation von NC-Bearbeitungsabläufen, Überprüfung von Kollisionen und Bewertung von Bearbeitungsqualität bis hin zur Schulung von Mitarbeitern. Die Vorteile der Simulation für die NC-Fräsbearbeitung haben sich bereits in vielen Anwendungsfällen erwiesen [3].

Trotzdem sind viele Beschränkungen in den verfügbaren NC-Simulationssystemen zu finden. Die zur Zeit verfügbaren NC-Simulationssysteme, wie Vericut, dCADE, Lightwork, Hypermill und WorkNC, bieten umfangreiche Methoden für die 3- und 5-achsige Simulation von Fräsbearbeitungsvorgängen an. Diese Systeme haben eine Gemeinsamkeit. Die Simulation wird auf Basis von Konturlinien oder Dreiecksnetzen und nicht auf einem Volumenmodell durchgeführt. Nachteilig wirkt sich auch die Tatsache aus, dass aufgrund der internen Repräsentation nur eine 2.5D-Visualisierung realisiert werden kann. Dies bedeutet, dass der Wechsel des Betrachterstandortes eine Neuberechnung der Visualisierungsdaten erforderlich macht.

Außerdem werden meist Nadel-, Voxel-, und Dixelmodelle in der NC-Simulation eingesetzt, während die Fräsbahn über Sweep-Operationen erzeugt wird [7, 31, 74]. Dabei bestimmt oft die Grafikleistung des Rechners, in welcher Detailtreue die Simulation stattfindet. Hinzu kommt noch, dass diese Systeme nicht alle fünfachsentauglich sind, da die über den Sturzwinkel zeitlich veränderliche oberflächenerzeugende Werkzeuggeometrie nicht über die Sweep-Operation dargestellt werden kann [75].

Während Voelcker und Hunt die Möglichkeit eines Einsatzes des Constructive Solid Geometry (CSG)-Modellierungssystems (PADL) zur NC-Simulation untersucht haben [76, 77], hat Fridshal bei General Dynamics das Solid-Modellierung-Paket (TIPS) zur NC-Simulation verwendet [78]. Der Zeitbedarf der NC-Simulation wächst bei der CSG-Methode proportional zur vierten Potenz der Anzahl der NC-Bahnen [77]. Außerdem werden mit der Materialabtragung die Gestaltungsvorgänge des Werkstücks wie die Boolesche Operationen immer komplizierter, da die ganzen Gestaltungsvorgänge im Geometriemodell bei der CSG-Methode gespeichert werden müssen.

Wang-Wang [40] beschreibt eine Methode zur analytischen Darstellung von Zerspanungsvolumen und wobei er annäherungsweise das Zerspanungsvolumen in einen Polyeder umwandelt. Mit Hilfe der Scan-line-Methode werden der angenäherte Polyeder in einen erweiterten Z-Buffer für Zerspanungsvolumen umgewandelt. Schließlich werden die Booleschen Operationen für NC-Simulation zwischen dem Z-Buffer für Zerspanungsvolumen und dem Z-Buffer für das Werkstück durchgeführt und im Bildraum abgebildet [40]. Mit der Hilfe der Wang-Wang-Methode kann die fünfachsiges NC-Simulation ausgeführt werden. Die Wang-Wang-Methode gehört eigentlich zur graphischen Lösung für NC-Simulation.

Van Hook [5] hat auch eine reine graphische Methode zur NC-Simulation entwickelt. Zuerst wird das Werkzeug durch Pixel-Image im voraus im Bildraum dargestellt. Die Booleschen Operationen lassen sich im Bildraum zwischen Werkzeug-Pixel-Image und Werkstück-Pixel-Image nach NC-

Bahnen schrittweise verwirklichen. Diese Methode hingegen beschränkt sich auf dreiachsige NC-Simulation, da sich die Achsrichtung des Werkzeugs nicht ändern darf. Andernfalls muss für jede Änderung der Achsrichtung des Werkzeugs ein neues Werkzeug-Pixel-Image berechnet werden, und dies wäre zeitaufwendig.

Chappel [79] beschreibt eine Point-Vector-Technik zur NC-Simulation. Die Oberfläche des Werkstücks wird durch selektierte Punkte auf dieser Oberfläche approximiert. An jedem Punkt entsteht eine Linie, die mit der Normalen an diesem Punkt auf der Oberfläche identisch ist. Die NC-Simulation wird durch das Abschneiden von diesen Linien durch die Werkzeugbewegung verwirklicht, und nicht durch die echten Booleschen Operationen. Diese können vorteilhaft zur Visualisierung der Flächenqualität in der Schlichtbearbeitung eingesetzt werden. Sie lassen jedoch die Beurteilung von Zwischenzuständen nicht zu.

Drysdale und Jerard befassen sich eingehend mit den diskreten Methoden zur Generierung von NC-Programmen und deren Simulation, Verifikation und Korrektur. Der Schwerpunkt hierbei ist die Diskretisierung der Oberfläche des Werkstücks in Set-Punkte beziehungsweise Dreiecke unter der Berücksichtigung der Diskretisierungs- und NC-Simulationsfehler. Die NC-Programmierung bis zu fünf Achsen lässt sich damit einfach durchführen. Die NC-Simulation ist lediglich auf drei Achsen beschränkt [56, 80, 81, 82].

Stifter stellt mit NC-SAVE ein System vor, in dem der Objektraum durch ein Dixelmodell abgebildet wird, und somit das simulierte Werkstück wirksam aktualisiert werden kann. Er beschreibt eine Methode zur Begrenzung der Abweichung zwischen dem Werkstück und dem entsprechenden Dixelmodell. Die NC-Simulationsfehler wie Unterschnittfehler und Restmaterialfehler werden jedoch dabei nicht berücksichtigt [34]. Für die Visualisierung des Dixelmodells wird hier die Strahlverfolgungsmethode eingesetzt, deren Richtung mit der Dixelrichtung identisch ist. Die Änderungen des Betrachterstandpunkts oder Transformationen des Dixelmodells erfordern eine erneute Aktualisierung des Werkstückmodells. Ebenfalls stellt Huang eine dixelbasierte Methode zur NC-Simulation und der Korrektur der NC-Programmierungsfehler vor. Die Visualisierung des Werkstücks wird hingegen im Werkstückraum durch die Konturdarstellung des Dixelmodells durchgeführt. Somit kann das simulierte Werkstück ohne eine zusätzliche Berechnung der Visualisierungsdaten dynamisch auf dem Bildschirm dargestellt werden [7]. Mit der dixelbasierten Methode wird die Richtung des Dixelmodells optimal nach der Form des Werkstücks festgelegt. Für die fünfachsigige NC-Simulation, besonders bei komplexer geformten Teilen wie bei Turbinenrädern, ist es schwer, die optimale Richtung zu finden.

Von Blackmore [50] wird auch ein Simulationssystem für mehrachsige NC-Bearbeitungsvorgänge vorgestellt. Der Schwerpunkt ist hier die Generierung von Zerspanungsvolumen, basierend auf dem Sweep-Envelope Differential Equation Algorithm, mit der die kritischen Kurven auf der Werkzeugoberfläche entlang der Richtung der NC-Bahnen berechnet werden können. Für Boolesche Operationen und Visualisierung des Werkstücks wird ein kommerzielles CAD System eingesetzt.

2.7 Anforderungen an die Werkstückaktualisierung

Für eine realistische NC-Simulation hat die Aktualisierung der Werkstückgeometrie zentrale Bedeutung. Da die existierenden Verfahren die Modellierung und Visualisierung des simulierten Werkstücks nicht ausreichend unterstützen, müssen neue Ansätze ausgearbeitet werden. Neue innovative Verfahren der rechnerunterstützten 3D-Modellierung und verstärkte Rechenleistungen ermöglichen eine realitätsnahe Simulation des Abtragungsvorganges spanender Fräsbearbeitungsprozesse von Werkstücken.

Für die Werkstückaktualisierung in der NC-Simulation werden die Anforderungen zusammenfassend wie folgt dargestellt:

- Die Komplexität des simulierten Werkstücks muss beliebig gestaltbar sein,
- die Werkstückmodellierung darf nicht übermäßig genau sein, jedoch sollte die NC-Simulationsgenauigkeit kontrollierbar und determinierbar sein,
- eine vollständige Repräsentation der zur Visualisierung und Manipulation nötigen Volumendaten im Hauptspeicher muss auch für große Volumenmodelle gegeben sein,
- ein schneller lesender und manipulierender Zugriff auf die Elemente des Volumenmodells muss gegeben sein,
- die Booleschen Operationen müssen einfach durchführbar sein,
- mit den zunehmenden Materialabtragungen sollte kein überproportional steigender Zeitaufwand in der Werkstückmodellierung verbunden sein,
- die Restmaterialien und Materialverletzungen der Sollteilgeometrie müssen ermittelt werden,
- das simulierte Werkstück muss im 3D-Objektraum mit indirektem Verfahren visualisiert werden können. Die Änderung des Betrachterstandpunkts oder die Transformation des

Werkstücks muss sich ohne zusätzliche Neuberechnungen der Visualisierungsdaten ausführen lassen,

- zur Verbesserung der Visualisierungsqualität muss die Oberfläche des simulierten Werkstücks extrahiert und gezeichnet werden können,
- zur echtzeitnahen NC-Simulation muss die Datenstruktur des Modellierkerns für NC-Simulation für die Anwendung der modernen Rechentechniken wie Parallelisierung geeignet sein.

3 Gestaltung des Voxelmodells zur NC-Simulation unter Beschränkung der Simulationsfehler

3.1 Virtual Clay Modelling System - VCM-System

In der vorliegenden Arbeit wird im Rahmen der fünfschichtigen NC-Simulation das im VCM-System verwendete Voxelmodell eingesetzt. Deswegen sollte in diesem Kapitel zuerst das VCM-System - das Virtual Clay Modelling System, kurz dargestellt werden. Danach wird das in dieser Arbeit verwendete Voxelmodell im Vordergrund stehen. Anschließend wird die Methode zur Begrenzung der Fehler bei der voxelbasierten NC-Simulation beschrieben. Zum Schluss wird ein Materialmodell zur fünfschichtigen NC-Simulation rechnerintern im VCM-System erzeugt.

Das Virtual Clay Modelling System ist ein Designwerkzeug, mit dem der Designer in Analogie zum konventionellen Designprozess beliebig geformte Modelle generieren kann [24, 83]. Die Werkzeuge zur Modellierung des virtuellen Modellbaumaterials entsprechen denen der realen Tonmodellierung, sogenannten True Sweep, sowie Schaber und Schablonen. In einem Werkzeug-Editor lassen sich diese starren Werkzeuge, basierend auf 2D-Splines, erstellen und modifizieren. Bei der Modellierung, dem Auf- und Abtragen von Material, werden verschiedene Interaktionstechniken angewendet. Die Werkzeuge lassen sich mittels Maus, Spacemouse und Trackingsystem bedienen. Dadurch wird ein freies Modellieren in allen Freiheitsgraden möglich gemacht. Im Gegensatz zum traditionellen

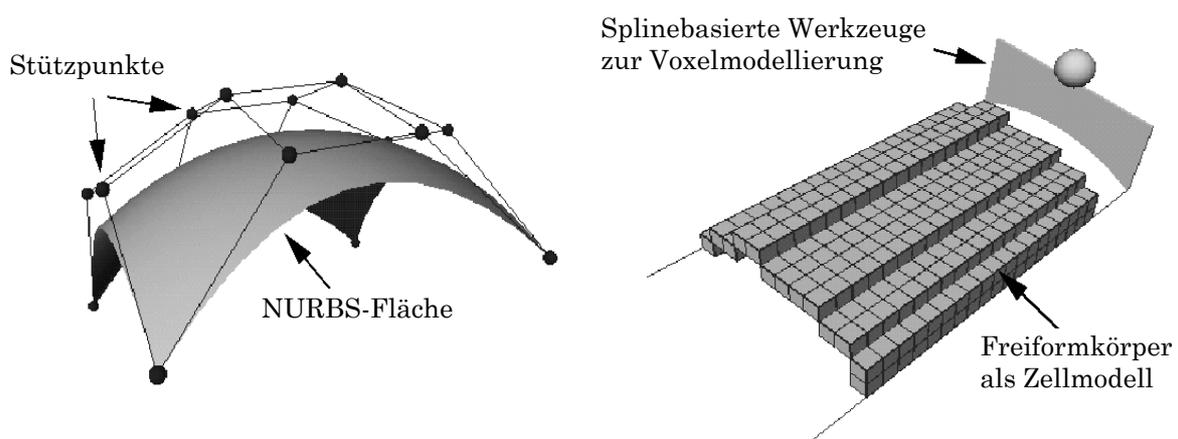


Bild 3.1: Prinzip des Virtual Clay Modellers [24]

Designsystem baut die Methode der virtuellen Tonmodellierung auf einem geänderten Paradigma auf. Im Vergleich zur splinebasierten Flächenmodellierung werden hier die Werkzeuge exakt durch

Splines repräsentiert, und die Modellierung erfolgt auf einem diskreten Volumenmodell, das durch Volumenelemente oder Voxeln dargestellt wird (Bild 3.1). Entsprechend dem realen Modellbau lässt sich das virtuelle Modellbaumaterial auftragen und abtragen. Diese Modellieroperationen werden auf elementare Boolesche Operationen zurückgeführt, welche auf dem Voxelmodell abgebildet werden. Die Modifizierung des Modells erfolgt in Abhängigkeit von der Bewegung des Werkzeugs. Das modifizierte Volumen oder auch Bewegungsvolumen wird in mehreren Zeitschritten bezüglich der Werkzeugposition ermittelt, und anschließend mittels Boolescher Operationen mit dem Ausgangsvolumen verknüpft.

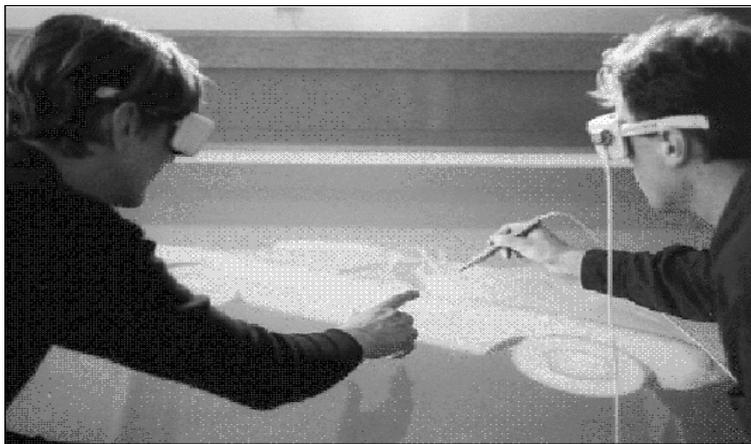


Bild 3.2: Einsatz einer Responsive Workbench[84]

In Verbindung mit Virtual-Reality(VR)-Techniken wird eine stereoskopische Darstellung, eine dreidimensionale Wahrnehmung möglich gemacht. Bild 3.2 stellt den Einsatz einer Responsive Workbench dar, die neben der Desktop-VR-Lösung ein intuitives Arbeiten in einer der realen Arbeitsumgebung entsprechenden Arbeitswelt ermöglicht.

Die Visualisierung des Modells wird mit dem auf die Datenstruktur des Modelliererkerns angepassten Marching-Cubes-Verfahren [66] durchgeführt. Dieses ermittelt die genauen Positionen der Dreiecke, die das Volumenmodell umhüllen. Zur Reduzierung der Anzahl der generierten Dreiecke und damit zur Beschleunigung der Visualisierung wird ebenso das adaptive Marching-Cubes-Verfahren [85] eingesetzt.

Die Qualität des Modells wird mit Hilfe von Highlight-Linien, einer Sonderform der Reflexionslinien, überprüft [86]. Beispielsweise sind damit Unstetigkeiten in den erzeugten Freiformflächen identifizierbar. Die Schichteneigenschaften des Voxelmodells ermöglicht eine direkte Kopplung der virtuellen Tonmodellierung mit Rapid-Prototyping-Anlagen über die Schnittstellen STL und CLI.

Mittels Stereolithographie und Laser-Sinter-Verfahren lassen sich schnell und einfach physische Modelle zur Evaluierung des Designs herstellen.

3.2 Das Voxelmodell im VCM-System²

3.2.1 Komprimierte Voxeldarstellung durch Stixel-Repräsentation

Im Rahmen der vorliegenden Arbeit wurde das Voxelmodell im VCM-System zur rechnerinternen Repräsentation der simulierten Werkstück- und Zerspanungsvolumen eingesetzt. In diesem Unterkapitel wird dessen Datenstruktur zugunsten der NC-Simulation dargestellt.

Bei Zelldarstellung wird der dreidimensionale Raum in lauter kubische Zellen von einer bestimmten Größe zerlegt. Dazu wird ein gleichmäßiges Gitter von parallel zu den Koordinatenachsen verlaufenden Strahlen in die Szene gelegt. In der klassischen Zelldefinition (Voxeldefinition) lassen sich nur die Schnittpunkte dieser Gitterlinien darstellen. Der Materialwert jedes Voxels lässt sich mit dem Material des Volumenkörpers, zu dem dieser Punkt gehört (Bild 2.2a), identifizieren. In der erweiterten Voxel mit Residuen-Darstellung können jedem Voxel der Berandung zusätzliche Werte zugeordnet werden. Diese Werte stellen den genauen Abstand des Schnittpunkts zur Oberfläche des Volumens entlang einer Gitterlinie dar (Bild 2.3a). Jedem Voxel lassen sich maximal sechs verschiedene Residuen zuordnen, da er sich am Schnittpunkt von drei Gitterlinien befindet und die Residuenwerte in beide Richtungen entlang einer Gitterlinie verlaufen können. In der Stixeldarstellung wird völlig auf die Ermittlung von Voxeln bzw. Residuen verzichtet. Alle Schnittpunkte der Oberflächen der Volumenkörper im Raum, die auf einer Gitterlinie liegen, werden direkt in einem Stixel repräsentiert.

Ein Stixel stellt alle Schnittpunkte, die durch eine Gitterlinie des Voxelraums und die Oberflächen der im Raum enthaltenen Volumenkörper gebildet werden, in einer Liste dar.

Speicherplatzbedarf

Für die verschiedenen Datenstrukturen ergeben sich die folgenden Formeln für den Speicherplatzbedarf:

$$M_{Matrix} = n^3 * M_{Material} \quad (3-1)$$

² Das vorgestellte Konzept wurde von Herrn cand.-ing. Arne Gerner entwickelt.

$$M_{Stixel} = n^2 * w * (M_{Material} + M_{Position} + M_{Liste}) \quad (3-2)$$

$$M_{Stick_Resid} = M_{Stick} + 3 * n^2 * w * (M_{Resid} + M_{Position} + M_{Liste}) \quad (3-3)$$

n ist die durch die Anzahl der Voxel dargestellte Kantenlänge des Voxelraums, M gibt den Speicherplatzbedarf an und w ist die durchschnittliche Anzahl von Materialwechslern entlang einer für die Lauflängenkodierung benutzten Voxelreihe. Besonders hervorzuheben ist hierbei die Ordnung von M_{Matrix} $O(n^3)$, M_{Stick} und M_{Stick_Resid} nur $O(n^2)$. Der Speicherplatzbedarf wächst bei der dreidimensionalen Matrix kubisch und bei den Stixel-Darstellungen lediglich quadratisch mit der Kantenlänge des Voxelraums.

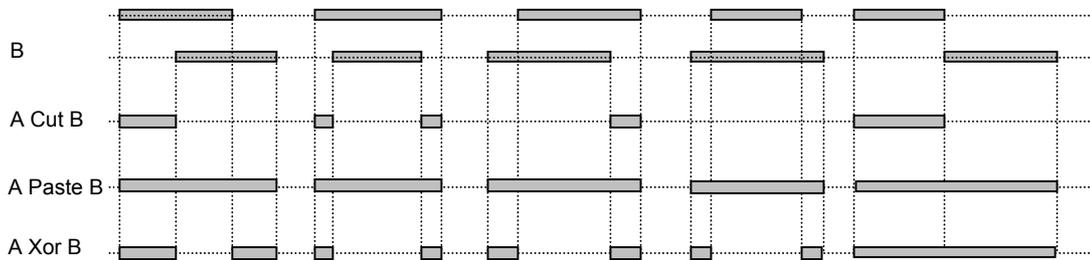


Bild 3.3: Ablauf verschiedener Mischoperationen

A	B	A Cut B	A Paste B	A Xor B
Luft	Luft	Luft	Luft	Luft
Luft	Ton	Luft	Ton	Ton
Ton	Luft	Ton	Ton	Ton
Ton	Ton	Luft	Ton	Luft

Tabelle 3.1: Materialkombinationen bei verschiedenen Mischoperationen

3.2.2 Die Mischoperationen

Die Mischoperationen werden zur Ermittlung der resultierenden Materialwerte der Elemente verschiedener Volumenmodelle definiert und einfach anhand von Tabellen, wie Tabelle 3.1, dargestellt. Diese Mischoperationen können weitgehend durch die aus der Aussagenalgebra bekannten Booleschen Operationen dargestellt werden. In Tabelle 3.1 und Bild 3.3 wird anhand der Kombination von zwei Materialien A und B, die den Wert Luft und Ton annehmen können, die Wirkung der drei

Mischoperationen Cut, Paste und Xor aufgezeigt. Mit den Mitteln der Aussagenalgebra ergibt sich aus Tabelle 3.1 für die Mischoperationen:

$$APasteB = A \vee B \quad (3-4)$$

$$ACutB = A \wedge \overline{B} \quad (3-5)$$

$$AXorB = \overline{A \equiv B} = A \neq B \text{ (Antivalenz)} \quad (3-6)$$

Da die Implikation als $A \supset B = \overline{A} \vee B$ definiert ist, kann Cut auch als negierte Implikation ausgedrückt werden.

$$ACutB = A \wedge \overline{B} = \overline{\overline{A \wedge \overline{B}}} = \overline{\overline{A} \vee B} = \overline{\overline{A} \vee B} = \overline{A \supset B} = A \not\supset B \quad (3-7)$$

Die Aussagenalgebra stellt eine binäre Boolesche Algebra dar. Bei der Definition von mehr als zwei Materialwerten kann die Aussagenalgebra nicht direkt angewendet werden, da sie nur für binäre, also zweielementige Mengen definiert ist. Werden neben den zwei Materialwerten Luft und Ton noch weitere Materialien wie beispielsweise Granit und Glas dargestellt, muss die Definition der Mischoperationen entsprechend erweitert werden.

3.2.3 Datenstruktur

In Bild 3.4 wird die Datenstruktur des Modellierkerns des VCM-Systems gezeigt. Das grundlegende Element der Datenstruktur ist der Stixel. In einem Stixel werden alle Schnittpunkte einer Gitterlinie des Voxelraums mit den Oberflächen der im Raum enthaltenen Volumenkörper in einer Liste von Materialwechseln gespeichert. Die Materialwechsel lassen sich durch die Abfrage der durch den Stixel repräsentierte Gitterlinie in positiver Richtung auf Schnittpunkte mit Volumenkörpern im Raum gewinnen. Die Position jedes dieser Schnittpunkte auf der Gitterlinie wird zusammen mit dem Material, das im Voxelraum in Richtung der Abfrage hinter dem Schnittpunkt liegt, in einem Materialwechsel als Element der Liste, die den Stixel darstellt, abgelegt. Es können noch beliebige Zusatzinformationen in einem Materialwechsel gespeichert werden, insbesondere sind dies die Oberflächennormalen. Alle Stixel einer Raumrichtung lassen sich in einer zweidimensionalen Matrix zusammenfassen. Ein vollständiger Voxelraum wird daher durch drei diese Matrizen in der X-, Y- und Z-Richtung dargestellt. Auf der Zugriffsebene des Voxelraumes wird mit globalen Koordinatenwerten gearbeitet, so dass Offsets, Koordinatenvertauschungen und sonstige Besonderheiten der internen Datenverarbeitung nach außen hin transparent gehandhabt werden können.

Zur Gewährleistung einer einheitlichen Behandlung der Matrizen für verschiedene Raumrichtungen werden bei äußeren Zugriffen auf die Stixel in der Matrix die Koordinaten des Koordinatensystems des Voxelraums in das Koordinatensystem der Matrix umgewandelt. Diese Umwandlung erfolgt mittels einer einfachen Vertauschung der Koordinatenachsen. Innerhalb jeder Matrix geben die X- und Y-Koordinaten die Position eines Stixels in der Matrix an, und die Z-Achse stellt immer die „aktive“ Richtung des Stixels dar.

Bei einer Umwandlung von äußeren in innere Koordinaten werden die Koordinaten also immer so vertauscht, dass die „aktive“ Achse der Matrix zur Z-Achse wird. Die beiden übrigen Koordinaten lassen sich im Prinzip beliebig behandeln. Aus Gründen der Datenkonsistenz lässt sich aber die Relation zueinander beibehalten (Tabelle 3.2).

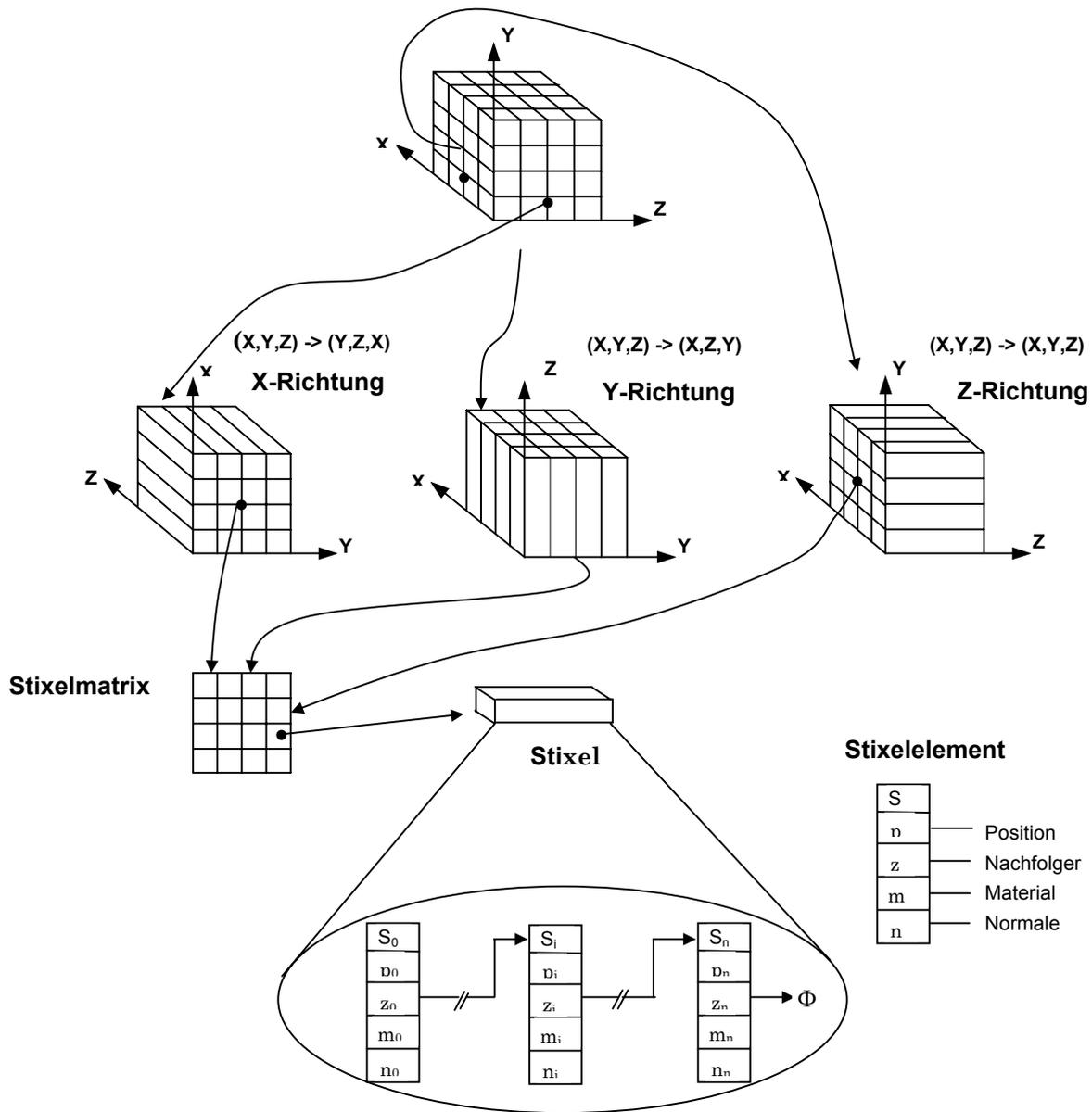


Bild 3.4: Datenstruktur des Modellierkerns des VCM-Systems

Aktive Achse	Vertauschung	
	Extern -> intern	Intern -> extern
X	$(x, y, z) \rightarrow (y, z, x)$	$(x, y, z) \rightarrow (z, x, y)$
Y	$(x, y, z) \rightarrow (x, z, y)$	$(x, y, z) \rightarrow (x, z, y)$
Z	$(x, y, z) \rightarrow (x, y, z)$	$(x, y, z) \rightarrow (x, y, z)$

Tabelle 3.2: Koordinatenvertauschungen für die verschiedenen Achsen

3.3 Marching-Cubes-Verfahren zur Visualisierung des Voxelmodells

Als Basis für die Visualisierung des Voxelmodells im VCM-System wird das Marching-Cubes-Verfahren [67] ausgewählt. Das Marching-Cubes-Verfahren wird in zwei Schritte eingeteilt. Im ersten Schritt lassen sich die Materialwerte von acht benachbarten Eckpunkten (Bild 3.5) festlegen. Dabei wird diese Materialverteilung in einer 8-Bit-Variable zwischengespeichert. Jedes Bit der Variable entspricht dem Wert eines Eckpunktes. Diese 8-Bit-Variable wird dann im zweiten Schritt als Index zur Festlegung der Position und der Lage der Dreiecke eingesetzt. Überschreitet (oder unterschreitet) der Wert eines Eckpunktes einen gegebenen Schwellwert – befindet er sich also innerhalb (oder außerhalb) des zu approximierenden Objektes - wird das entsprechende Bit mit dem binären Wert „1“ (oder „0“) belegt.

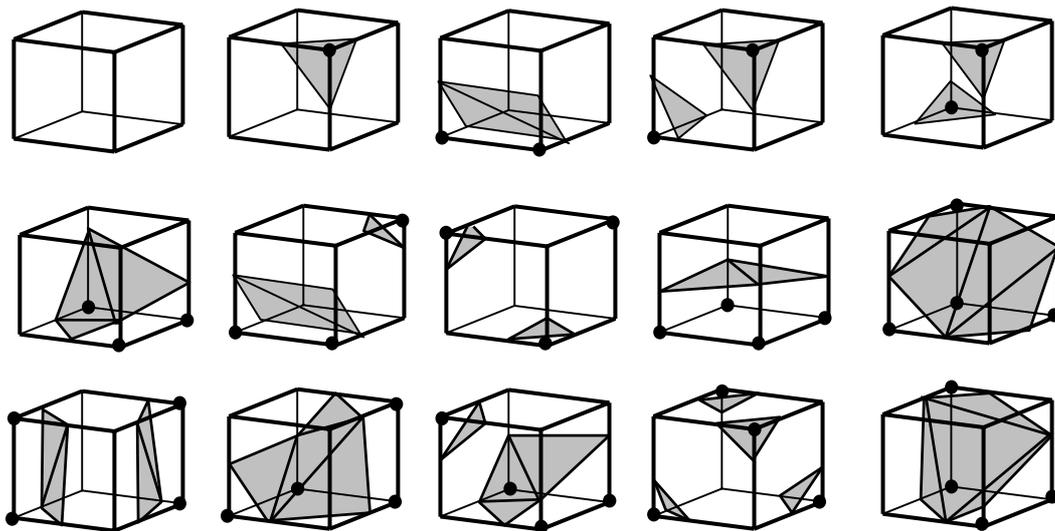


Bild 3.5: Die 14 Fälle des Marching-Cubes [67]

Da jede Zelle acht Eckpunkte hat und jeder Eckpunkt möglicherweise zwei Statusse, nämlich innerhalb oder außerhalb des Objektes, besitzt, ergeben sich insgesamt für eine Zelle 2^8 , also 256 verschiedene Konfigurationen. Zur Speicherung dieser 256 Konfigurationen von Dreiecken könnte eine Tabelle eingesetzt werden. Für den Aufbau dieser Tabelle reichen jedoch bereits 14 Konfigurationen aus, da alle übrigen Möglichkeiten aus den 14 Grundkonfigurationen durch Rotation, Spiegelung oder Invertierung abgeleitet werden können (Bild 3.5).

Die Oberflächenextraktion des Voxelmodells erfolgt nun sequentiell in X-, Y- und Z-Richtung durch den Volumen-Datensatz. Werden die Materialwerte einer Zelle festgelegt, erfolgt dann die Triangu-

lation dieser Zelle. Der Marching-Cubes ist in der Lage, die Zelle unabhängig voneinander zu berechnen. Jedoch entsteht immer ein geschlossenes Modell, da die Schnittpunkte des Objektes mit zwei benachbarten Zellen an der gemeinsamen Fläche identisch sind und somit ein kontinuierlicher Übergang zwischen den Zellen existiert³.

Nach der Oberflächenextraktion des Voxelmodells erfolgt direkt die Visualisierung der Datenstrukturen mit der Graphikbibliothek Open Inventor, welche auf die GL-Graphikbibliothek aufsetzt und in hohem Maße auf die Graphik-Hardware zugeschnitten ist. Bild 2.1 zeigt ein durch Marching-Cubes-Verfahren erzeugtes Oberflächenmodell.

3.4 Fehlerbegrenzung in der voxelbasierten fünfsichtigen NC-Simulation

3.4.1 Begrenzung des Flächenfehlers in der voxelbasierten Geometriedarstellung

Im Stixel-Datenformat für die Repräsentation von dreidimensionalen Daten werden die Positionen des Materialwechsels gespeichert. Dazu werden von drei Ebenen $E_1(x, y)$, $E_2(y, z)$ und $E_3(z, x)$ senkrechte Strahlen in die Szene gelegt. Während entlang der Strahlen die Oberflächenpositionen mit Fließkommagenauigkeit gespeichert werden können, haben diese Strahlen zwangsläufig einen diskreten Abstand zueinander.

Bei der voxelbasierten fünfsichtigen NC-Simulation wird das Werkstück mit Stixeln und dessen Flächen mittels Sample-Punkten im 3D-Raum dargestellt. Zwischen den Sample-Punkten bestehen keine Informationen über diese Flächen. Es besteht auch die Möglichkeit, dass das simulierte Werkstück in jedem Sample-Punkt die vorgegebene Programmierungstoleranz nicht überschreitet. In anderen Orten zwischen den Sample-Punkten wird jedoch die jeweilige Fläche verletzt oder es werden darauf überflüssige Materialien hinterlassen. Um den Simulationsergebnissen gerecht werden zu können, wird der in der Simulation eingeführte sogenannte Simulationsfehler in Betracht gezogen.

Ausgehend von der Analyse des NC-Simulationsfehlers im Abschnitt 2.4 können die Simulationsfehler in Flächenfehler ε_f , Unterschnittfehler ε_u und Restmaterialfehler ε_r unterteilt werden (Bild

³ Das vorgestellte Konzept wurde von Herrn cand.-ing. Tapani Hegewald entwickelt.

2.6). Im folgenden werden die Simulationsfehler detailliert analysiert und die Methoden zur Kontrolle der NC-Simulationsfehler beschrieben.

Bei der diskreten Geometriedarstellung existiert zwischen dem Flächenfehler und der Anzahl der Sample-Punkte sowie den Krümmungen der Sollfläche eine direkte Abhängigkeit. Nimmt die Anzahl der Sample-Punkte innerhalb eines hoch gekrümmtes Bereiches zu, verringert sich der Flächenfehler in diesem Bereich. Die genaue Berechnung der Flächenfehler ist jedoch sehr umständlich [56]. Somit ist es erforderlich, eine konservative Methode zur Begrenzung der Flächenfehler einzusetzen.

Dabei wird davon ausgegangen, dass der maximale Flächenfehler normalerweise bei der Diskretisierung im Flächenbereich mit der höchsten Krümmung auftritt. Es wird angenommen, dass die Fläche im Bereich mit der höchsten Krümmung durch die entsprechende Schmiegekugel angenähert wird. Die Schmiegekugel ist die Kugel, die die Fläche an einem Punkt so berührt, dass die zweiten Ableitungen von Fläche und Kugel an dem Punkt identisch sind. Bild 3.6 zeigt eine Schmiegekugel. ε_f ist der maximale Flächenfehler. R ist der Schmiegekugelradius. Der Durchmesser (D) des Kreises, welche durch die Kreuzung zwischen der Schmiegekugel und einer Dreiecksfläche erzeugt wird, wird wie folgt bezeichnet:

$$D = 2\sqrt{2R\varepsilon_f - \varepsilon_f^2} \quad (3-8)$$

Da ε_f im Vergleich mit dem Schmiegekugelradius sehr klein ist, ergibt sich

$$D = 2\sqrt{2R\varepsilon_f} \quad (3-9)$$

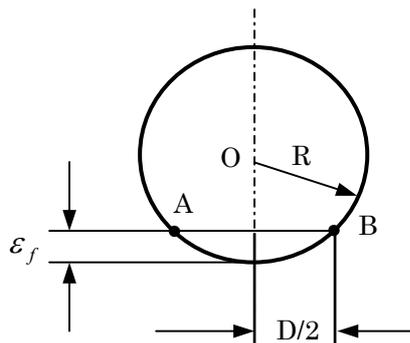


Bild 3.6: Die Seitenansicht der Kreuzung zwischen der Schmiegekugel und einer Ebene

Wenn die Schmiegekugel mit zwei Endpunkten der Diagonale eines Würfels (Bild 3.7) im Voxelmodell überlappt, tritt der größte Flächenfehler auf (siehe Anhang A).

Dann ergibt sich

$$H = \frac{2}{3} \sqrt{6R\varepsilon_f} \quad (3-10)$$

Falls die größte Krümmung der Fläche k_{\max} bekannt ist, lässt sich diese Formel in folgender Form beschreiben:

$$H = \frac{2}{3} \sqrt{\frac{6\varepsilon_f}{k_{\max}}} \quad (3-11)$$

In dieser Formel kann der Diskretisierungsgrad (H) des Voxelmodells durch die maximale Flächenkrümmung und den vom Konstrukteur vorgegebenen Flächenfehler zuerst bestimmt werden. Der Flächenfehler wird dadurch gewährleistet, dass das Werkstückmodell durch ein Stixelmodell mit der

Gittergröße $\frac{2}{3} \sqrt{\frac{6\varepsilon_f}{k_{\max}}}$ diskretisiert wird.

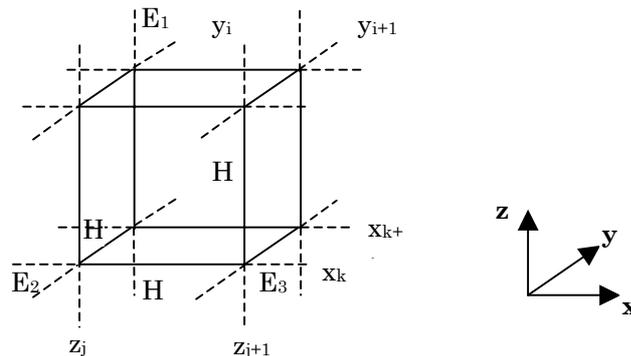


Bild 3.7: Ein Würfel im Voxelmodell

3.4.2 Begrenzung des Unterschnittfehlers in der voxelbasierten fünfachsigen NC-Simulation

In der voxelbasierten NC-Simulation wird die Oberfläche des Werkstücks durch die Dreiecksnetze angenähert. Es kann vorkommen, dass die Dreiecksfläche zwischen Sample-Punkten unterschritten wird, aber ohne Verletzungen der Oberfläche des Werkstücks in den Sample-Punkten. Dieser Unterschnitt hängt direkt mit dem Fräserdurchmesser und dem Abstand zwischen zwei Sample-Punkten zusammen. Falls sich der Mittelpunkt eines Kugelfräasers oberhalb der Dreiecksfläche befindet, kann die folgende Formel für die Begrenzung des Unterschnittfehlers wie 3.4.1 abgeleitet werden.

$$H = \frac{2}{3} \sqrt{6r_c \varepsilon_u} \quad (3-12)$$

r_c ist der Fräserradius.

Wenn der Mittelpunkt eines Kugelfräasers außerhalb der Dreiecksfläche liegt, tritt der maximale Unterschnittfehler bei einer Dreieckskante auf. In diesem Fall tritt eine seitliche Überschreitung in das Dreieck abc (Bild 3.8) auf. Die Überschreitung ε_1 lässt sich wie folgt ableiten.

$$\varepsilon_1 = \varepsilon_u \quad (3-13)$$

Es ist deutlich dargestellt, dass die seitliche Überschreitung den Unterschnittfehler ε_u nicht überschreitet. Die Formel (3-13) basiert auch auf der Annahme $\varepsilon_1 \ll r_c$.

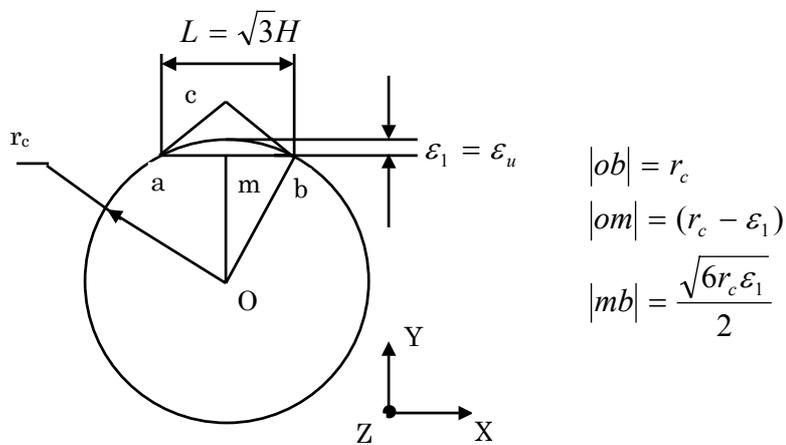


Bild 3.8: Ansicht eines Dreieckes und einer Kugel, deren Mittelpunkt außerhalb des Dreieckes liegt.

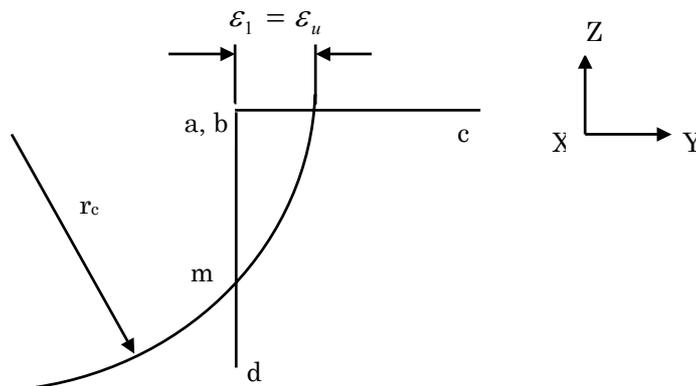


Bild 3.9: Seitenansicht der im Bild 3.8 gezeigten Kugel und Darstellung der Kante des Dreieckes abc und der angrenzenden Kante des Dreieckes abd

entsprechenden Dreieckes Materialien übrig. Die maximale Höhe der übrig gebliebenen Materialien auf dem Dreieck wird als Restmaterialfehler ε_r bezeichnet. Der Restmaterialfehler befindet sich normalerweise in der Mitte des jeweiligen Dreieckes. Es ist jedoch schwer, eine exakte Berechnung der Restmaterialien lediglich durch Untersuchung der Höhe von Sample-Punkten durchzuführen. Deshalb ist es auch erforderlich, eine konservative Methode zur Begrenzung des Restmaterialfehlers zu entwickeln.

Um diese konservative Methode zur Begrenzung des Restmaterialfehlers darzustellen, wird hier zuerst im 2D-Raum der maximale „Restmaterialfehler“ auf einem Kreis (mit Radius $R = \frac{1}{k_{\max}}$) angenähert durch ein Polygon untersucht. Es wird angenommen, dass ein Kreis O_2 durch ein Polygon A, A1, ..., A2 angenähert wird. Dieses Polygon lässt sich dann durch einen „2D-Fräser“ O_1 (Kreis) mit dem Radius r_c bearbeiten. Dabei ist ersichtlich, dass sich der „2D-Fräser“ O_1 immer denn weiter vom Mittelpunkt der Kante AA1 entfernt, wenn er sich stetig um den Punkt A in Richtung Punkt A2 dreht. Wenn der „2D-Fräser“ O_1 den Punkt A2 berührt, liegt der maximale „Unterschnittfehler“ innerhalb des Bereichs A-A2. Gleichzeitig bleiben die „maximalen Restmaterialien“ auf dem Mittelpunkt der Kante AA1 bestehen. Da der „maximale Restmaterialfehler“ begrenzt werden soll, bildet hier die Strecke AA2, AA1 die längste Kante des Polygons (Bild 3.10). In diesem Beispiel wird die längste Kante des Polygons als d bezeichnet. Der „maximale Restmaterialfehler“ ε_r wird nach folgender Formel ermittelt:

$$\sqrt{r_c^2 - \left(\frac{d}{2} + AB\right)^2} + \varepsilon_r = \sqrt{r_c^2 - AB^2} \quad (3-14)$$

mit

$$AB = r_c \sin(2\theta + \phi)$$

$$\sin 2\theta = \frac{d}{R^2} \sqrt{R^2 - \left(\frac{d}{2}\right)^2}$$

$$\cos 2\theta = 1 - \frac{d^2}{2R^2}$$

$$\sin \phi = \frac{d}{2r_c}$$

$$\cos \phi = 1 - \frac{d^2}{4r_c^2}$$

Daraus ergibt sich:

$$\frac{1}{4R^2}d^4 - \left(\frac{3}{4} + \frac{lm}{R^2} + \frac{\varepsilon_r l}{R^2} + \frac{\varepsilon_r m}{R^2}\right)d^2 + 2\varepsilon_r l - \varepsilon_r^2 = 0 \quad (3-15)$$

$$\text{mit } l = \sqrt{r_c^2 - \left(\frac{d}{2}\right)^2}, \quad m = \sqrt{R^2 - \left(\frac{d}{2}\right)^2}$$

Man erhält

$$d^2 = 2R^2 \left(\frac{3}{4} + \frac{lm}{R^2} + \frac{\varepsilon_r l}{R^2} + \frac{\varepsilon_r m}{R^2} - \sqrt{\left(\frac{3}{4} + \frac{lm}{R^2} + \frac{\varepsilon_r l}{R^2} + \frac{\varepsilon_r m}{R^2}\right)^2 - \frac{2\varepsilon_r l}{R^2} + \frac{\varepsilon_r^2}{R^2}} \right) \quad (3-16)$$

Um ein konservatives Ergebnis für d^2 zu erzielen, wird die Formel (3-16) weiter umgeformt.

$$d^2 = 2R^2 \left(\frac{3}{4} + \frac{lm}{R^2} + \frac{\varepsilon_r l}{R^2} + \frac{\varepsilon_r m}{R^2} - \sqrt{\left(\frac{3}{4} + \frac{lm}{R^2} + \frac{\varepsilon_r m}{R^2}\right)^2 + \frac{\varepsilon_r l}{R^2} \left(2 * \frac{l}{R} \frac{m}{R} + 2 * \frac{\varepsilon_r}{R} \frac{m}{R} - \frac{1}{2} + \frac{\varepsilon_r}{R} \frac{l}{R} + \frac{\varepsilon_r}{l}\right)} \right) \quad (3-17)$$

Es gilt allgemein für die NC-Programmierung: $l < R$, $m < R$, $\varepsilon_r \ll r_c$, $\varepsilon_r \ll R$

Es ist aus der Formel (3-17) zu ersehen, dass

$$\begin{aligned} \frac{3}{4} + \frac{lm}{R^2} + \frac{\varepsilon_r m}{R^2} &> \frac{3}{4} \\ -\frac{1}{2} &< 2 * \frac{l}{R} \frac{m}{R} + 2 * \frac{\varepsilon_r}{R} \frac{m}{R} - \frac{1}{2} + \frac{\varepsilon_r}{R} \frac{l}{R} + \frac{\varepsilon_r}{l} < \frac{3}{2} \\ \frac{\varepsilon_r l}{R^2} &\ll 1.0 \end{aligned}$$

somit wird abgeleitet:

$$d^2 \approx 2R^2 \left(\frac{3}{4} + \frac{lm}{R^2} + \frac{\varepsilon_r l}{R^2} + \frac{\varepsilon_r m}{R^2} - \sqrt{\left(\frac{3}{4} + \frac{lm}{R^2} + \frac{\varepsilon_r m}{R^2} \right)^2} \right) = 2\varepsilon_r l \approx 2\varepsilon_r r_c \quad (3-18)$$

Daraus ergibt sich:

$$d \approx \sqrt{2r_c \varepsilon_r} \quad (3-19)$$

Bei der voxelbasierten NC-Simulation wird die Oberfläche des Werkstücks durch Dreiecksnetze angenähert. Es wird angenommen, dass eine Kugel (vereinfachter Fräser) zuerst auf einem Eckpunkt eines Dreieckes positioniert ist. Sie hat hierbei gewisse Abstände zu den anderen Eckpunkten der sich um den Eckpunkt befindenden Dreiecke. In diesem Fall verbleiben die Restmaterialien auf allen Dreiecken. Dreht sich die Kugel stetig um den Eckpunkt nach einer Richtung, entfernt sie sich immer weiter von den sich gegen diese Richtung befindenden Dreiecken. Dreht sich die Kugel nach einer bestimmten Richtung um diesen Eckpunkt solange, bis sie die anderen zwei Eckpunkte eines Dreieckes berührt, tritt der maximale Unterschnittfehler unterhalb des Dreieckes auf. Gleichzeitig bleiben die maximalen Restmaterialien auf den sich gegen diese bestimmte Richtung befindenden Dreiecken. Es zeigt sich, dass bei einer Überlappung der Kugel mit drei Eckpunkten (E_1, E_2, E_3) eines Würfels (Bild 3.7) im Voxelmodell der größte Unterschnittfehler unterhalb des Dreieckes $E_1E_2E_3$ auftritt (siehe Anhang A). Da der maximale Restmaterialfehler in der voxelbasierten NC-Simulation begrenzt werden soll, lässt sich die maximale Kante d in der Formel (3-19) wie folgt ermitteln: $d = \sqrt{3}H$. H ist die Kantenlänge eines Würfels des Voxelmodells.

Es ergibt sich dann aus der Formel (3-19):

$$H = \sqrt{\frac{2}{3} r_c \varepsilon_r} \quad (3-20)$$

In dieser Formel kann der Diskretisierungsgrad (H) des Voxelmodells durch den Fräserradius und den vom Konstrukteur vorgegebenen Restmaterialfehler zuerst bestimmt werden. Der Restmaterialfehler wird dadurch gewährleistet, dass das Werkstückmodell mittels Stixel-Repräsentation mit

Gittergröße $\sqrt{\frac{2}{3} r_c \varepsilon_r}$ diskretisiert wird.

3.4.4 Zusammenfassung

Die NC-Simulationsfehler sind nicht an jeder Stelle des Werkstücks konstant. Dies ist darin begründet, dass zum einen sich die einzelnen Fehlerquellen nicht überall gleich auswirken und zum anderen die Fehlerwechselwirkungen an verschiedenen Raumpositionen den Gesamtfehler weiter erhöhen aber auch mindern können. Ausgehend davon, dass der NC-Simulationsfehler begrenzt werden muss, werden die maximalen Abweichungen zwischen der Sollfläche und der gefrästen Fläche ermittelt. Diese maximalen Abweichungen werden in zwei Kategorien gegliedert. Die erste maximale Abweichung ε_i entsteht, wenn sich ein gefräster Punkt im inneren Sollwerkstück befindet, und die Sample-Punkte exakt geschnitten werden. Die andere maximale Abweichung ε_o entsteht, wenn sich ein gefräster Punkt im äußeren Sollwerkstück befindet, und die Sample-Punkten exakt geschnitten werden.

Wie im Bild 3.11 gezeigt wird, gibt es für die erste maximale Abweichung ε_i :

$$\varepsilon_i = \varepsilon_u + \varepsilon_f \quad \text{für konvexe Flächen} \quad (3-21)$$

$$\varepsilon_i = |\varepsilon_u - \varepsilon_f| \quad \text{für konkave Flächen} \quad (3-22)$$

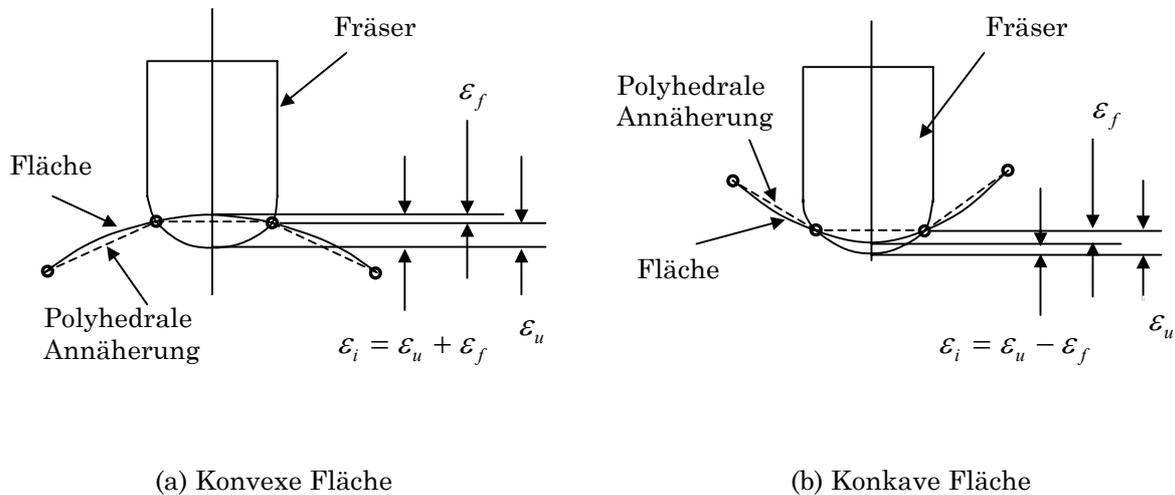


Bild 3.11: Flächenfehler und Unterschnittfehler in der NC-Simulation[56]

Für die zweite maximale Abweichung gibt es ähnliche Ergebnisse:

$$\varepsilon_o = \varepsilon_r + \varepsilon_f \quad \text{für konkave Flächen} \quad (3-23)$$

$$\varepsilon_o = |\varepsilon_r - \varepsilon_f| \quad \text{für konvexe Flächen} \quad (3-24)$$

Da die Flächenfehler normalerweise im Vergleich zu Unterschnitt- und Restmaterialfehler geringfügig sind, basiert der anfängliche Diskretisierungsgrad nur auf dem Unterschnitt- und Restmaterialfehler. Normalerweise kann dadurch die innere Abweichung ausreichend begrenzt werden [82, 56]. Beim Auftreten eines Flächenbereichs mit großer Krümmung wird der Flächenfehler zur weiteren Überprüfung der NC-Simulationsfehler eingesetzt.

Zusammenfassend lassen sich folgende Festlegungen treffen:

- Gewährleistung des Restmaterialfehlers. Beim Vergleich der Formel (3-12) mit der Formel (3-20) wurde festgestellt, dass der Diskretisierungsgrad in der Formel (3-20) (nach Restmaterialfehler) genau der Hälfte des Diskretisierungsgrades in der Formel (3-12) (nach Unterschnittfehler) entspricht. Dies bedeutet, dass der Unterschnittfehler auch dann gewährleistet ist, wenn der Diskretisierungsgrad nach der Formel (3-20) (nach Restmaterialfehler) am Anfang bestimmt wird. In diesem Fall gilt folgender Diskretisierungsgrad für die Definition des Stixelmodells in der voxelbasierten NC-Simulation: $H = \sqrt{\frac{2}{3}} r_c \varepsilon$. ε ist die vom Konstrukteur vorgegebene Toleranz für NC-Simulationsfehler. r_c ist der Fräserradius. H ist die Größe eines Würfels im Stixelmodell.
- Gewährleistung des Unterschnittfehlers. Ein Grundkriterium zur Programmierung für die fünfachsige NC-Fräsbearbeitung ist die Vermeidung der Unterschnitte vom Werkstück. Zur Verringerung der Anforderung des Rechnerspeichers in der voxelbasierten NC-Simulation lassen sich somit am Anfang nur die Unterschnittfehler begrenzen. Dadurch kann der Speicher für das Stixelmodell mehrfach reduziert werden. In diesem Fall gilt: $H = 2\sqrt{\frac{2}{3}} r_c \varepsilon$.
- Überprüfung mit Flächenfehler. In den vorherigen zwei Situationen wird der Flächenfehler bei der Begrenzung der NC-Simulationsfehler nicht herangezogen. Falls die Oberfläche eines Werkstücks stark gekrümmte Bereiche enthält, muss der NC-Simulationsfehler mittels Flächenfehler nachgeprüft werden. Überschreitet der NC-Simulationsfehler die Toleranz, verringert sich der Diskretisierungsgrad.

Tabelle 3.3 zeigt die Gittergröße (H) des Voxelmodells für Kugelfräser und Torusfräser mit unterschiedlichen Halbdurchmessern bei verschiedenen Unterschnittfehlern in der NC-Simulation. Beim Vergleich mit dem Dexelmodell zur NC-Simulation [34] bedarf die voxelbasierte Methode in der vorliegenden Arbeit weniger Rechnerspeicher.

Werkzeuge		H	ϵ_u				
			0.01	0.05	0.1	0.5	1.0
Kugel- fräser	R=4.0	0.33	0.73	1.03	2.31	3.27	
	R=6.0	0.40	0.89	1.26	2.83	4.00	
Torus- fräser	R1=4.0 R2=2.0	0.23	0.52	0.73	1.63	2.31	
	R1=6.0 R2=3.0	0.28	0.63	0.89	2.00	2.83	

**Tabelle 3.3: Vergleich der Gittergröße bei unterschiedlichen Werkzeugen und
Unterschnittfehlern**

3.5 Erstellung eines Voxelmaterialemodells mit dem VCM-System

Eine Voraussetzung zur voxelbasierten NC-Simulation ist die Modellierung eines Voxelmaterialemodells des Werkstücks. Im Rahmen der vorliegenden Arbeit wurde das voxelbasierte Materialemodell für NC-Simulation im VCM-System generiert. Als ein aussagekräftiges Beispiel zur fünfsichtigen NC-Simulation wurde ein Verdichterlaufrad ausgewählt. Im folgenden sind die Konzepte zur Modellierung eines Materialemodells des Verdichterlaufrades im VCM-System dargestellt. Die detaillierten Maße des Verdichterlaufrades werden im Kapitel 7 gegeben.

Es werden drei Operationsschritte im VCM-System zur Generierung des Materialemodells durchgeführt (Bild 3.12).

- Modellierung eines Voxelwürfels. Ausgehend von den Maßen des Verdichterlaufrades wurde die Größe des Würfels mit den Maßen 160.0 x 160.0 x 65.0 (mm) festgelegt (siehe Kapitel 7). Für die NC-Bearbeitung wurde ein Kugelfräser mit Halbdurchmesser 4.0 mm eingesetzt. Der Unterschnittfehler lässt sich unter 0.05mm begrenzen. Nach der Formel (3-12) wird die Voxelgröße mit 0.5 mm ausgewählt.
- Abtragung des Materials nach dem größten Durchmesser des Verdichterlaufrades. Ein einfacher Schaber wurde nach dem größten Durchmesser des Verdichterlaufrades entsprechend der Zentrallinie des Voxelwürfels positioniert, und um 360° gedreht. Die überflüssigen Materialien des Voxelwürfels wurden abgetragen. Ein Zylinder wurde erzeugt.

- Abtragung des Materials nach Außenprofil der Schaufel. Eine Schablone wurde gemäß dem Außenprofil der Schaufel mittels Werkzeugeditoren im VCM-System konstruiert. Danach wurde die Schablone entsprechend der Zentrallinie positioniert, und um 360° gedreht. Das Materialmodell im VCM-System wurde für weitere fünfschichtige NC-Simulationen generiert.

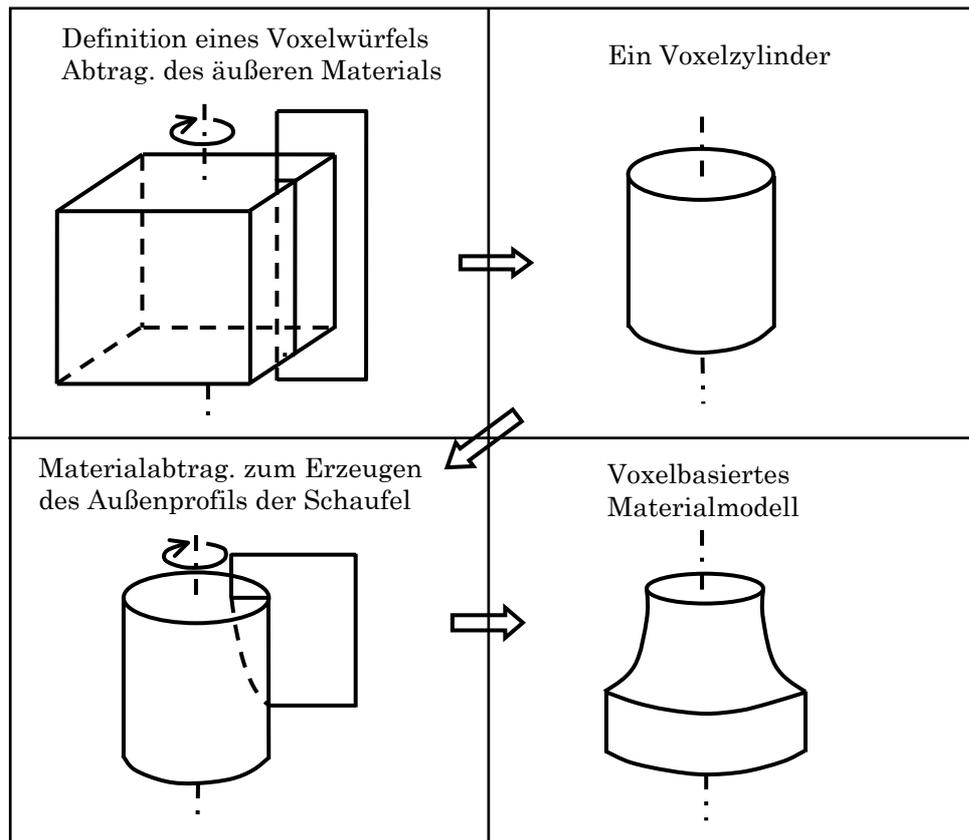


Bild 3.12: Vorgehensweise zur Modellierung des Materialmodells für die fünfschichtige NC-Simulation im VCM-System (Verdichterlaufrad)

Das im VCM-System erzeugte voxelbasierte Materialmodell lässt sich mit Stixel-Daten repräsentieren und exportieren. Das fünfschichtige NC-Simulationssystem hat dieses Modell übernommen. Bild 3.13 stellt das Materialmodell eines Verdichterlaufrades im NC-Simulationssystem dar.

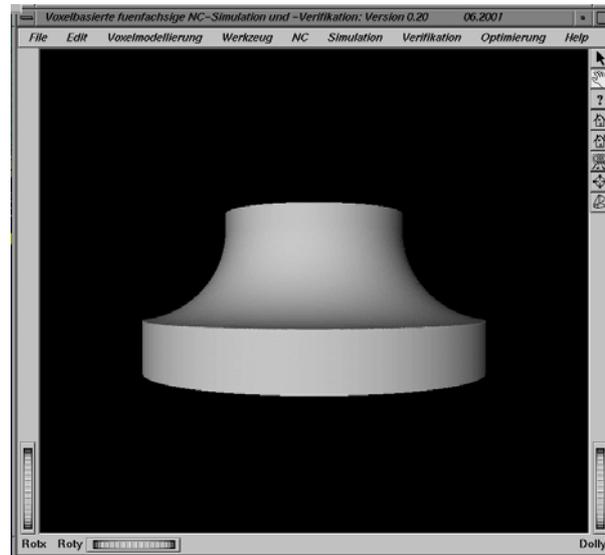


Bild 3.13: Ein voxelbasiertes Materialmodell in der fünfachsig NC-Simulation

4 Voxelbasierte Gestaltung des Zerspanungsvolumens in der NC-Simulation

4.1 Konzept zur Gestaltung des Zerspanungsvolumens in der voxelbasierten NC-Simulation

Ein wesentlicher Schritt zur voxelbasierten fünfsichtigen NC-Simulation ist die Modellierung des Zerspanungsvolumens im 3D-Raum. In diesem Kapitel wird zuerst ein Konzept zur Modellierung des Zerspanungsvolumens dargestellt. Anschließend werden die mathematischen Algorithmen zur Berechnung der oberflächenbasierten Zerspanungsvolumen diverser Werkzeuge beschrieben. Zum Schluss werden die berechneten Zerspanungsvolumen in voxelbasierte Zerspanungsvolumen umgewandelt, um daraus die voxelbasierte Materialabtragung ableiten zu können.

4.1.1 Leitgedanken zur Konzeption

Für die Simulation der Materialabtragung des Werkstücks muss das Zerspanungsvolumen des Werkzeugs zuerst geometrisch modelliert werden. Unter dem Zerspanungsvolumen versteht man den räumlichen Bereich, den das Werkzeug innerhalb eines Bewegungsintervalls (entsprechend einem NC-Satz) im 3D-Raum durchlaufen hat. Der durchfahrene Volumenbereich wird daher durch die Geometrie und die Bewegungsfunktion des Werkzeugs definiert.

Der Grundgedanke für die voxelbasierte Modellierung des Zerspanungsvolumens besteht darin, dass sich ein komplizierter Körper aus vielen kleinen und einfachen Grundkörpern zusammensetzt. Analog dazu kann man sagen, dass ein komplexes 3D-Zerspanungsvolumen aus mehreren einfachen Grundkörpern und deren Bewegungsvolumen aufgebaut werden kann. Die Grundkörper sind mit Hilfe weniger Parameter exakt zu beschreiben. Bild 4.1 stellt beispielsweise die Komponenten des Zerspanungsvolumens eines zylindrischen Gesenkfräasers dar.

Das Zerspanungsvolumen des Werkzeugs wird innerhalb eines Bewegungsintervalls in zwei Kategorien eingeteilt. Die eine entspricht dem Volumen des Werkzeugs in Anfangs- und Endposition des Bewegungsintervalls, die andere dem von der Außenkante des Werkzeugs innerhalb des Intervalls durchstrichenen 3D-Raum. Während die erste Kategorie von Volumen weiter in die zwei Grundkörper, Zylinder und Kugel unterteilt werden kann, setzt sich die zweite Kategorie aus den Bewegungsvolumen der jeweiligen Grundkörper zusammen. Die Grundkörper lassen sich durch ihre Geometrie und ihre räumliche Lage eindeutig festlegen. Hingegen werden die Bewegungsvolumen der Grundkörper durch die Zwischenpositionen der kritischen Kanten auf dem Grundkörper dargestellt [40,

87]. Im Rahmen der vorliegenden Arbeit wird die zweite Kategorie als Bewegungsvolumen bezeichnet.

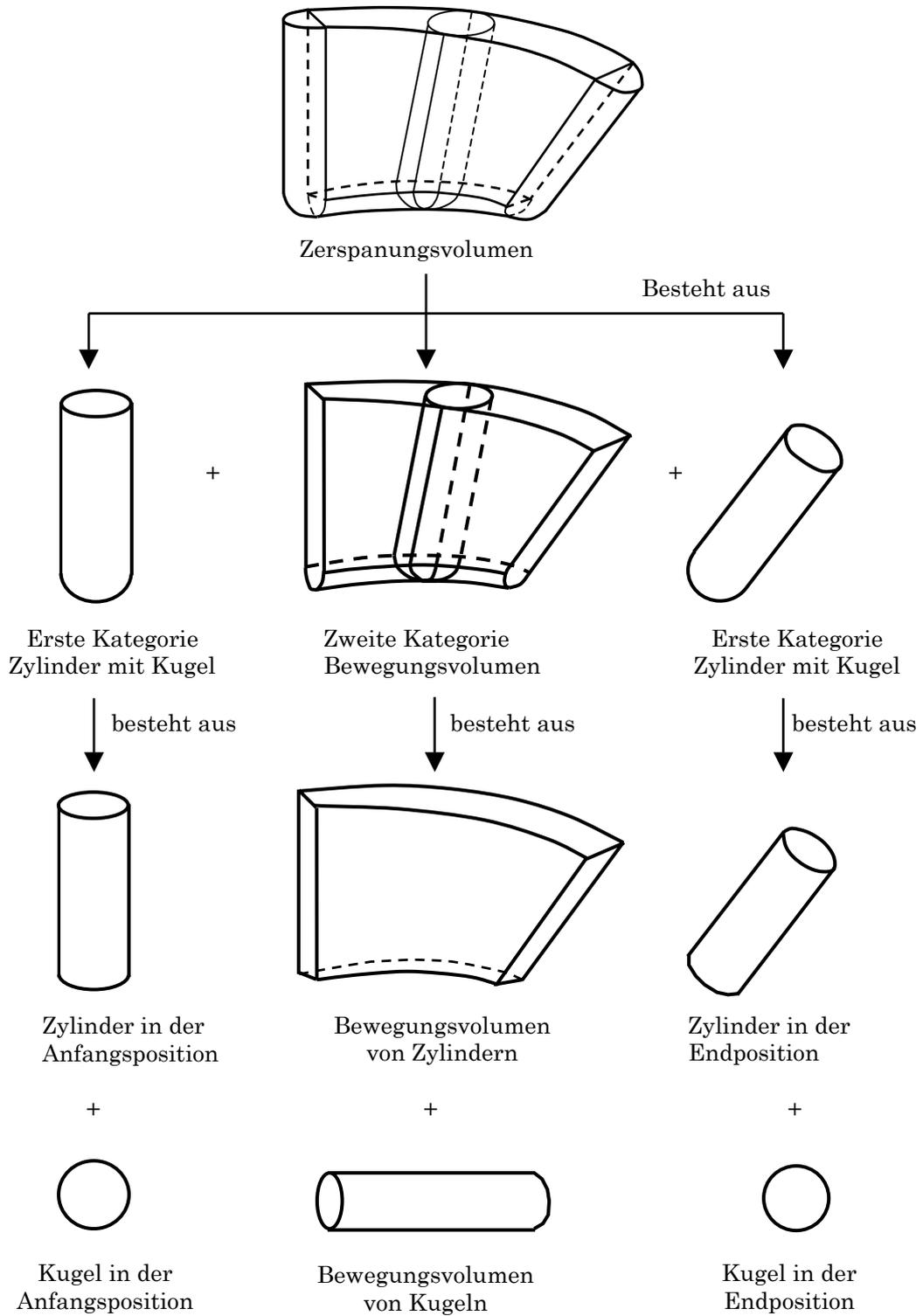


Bild 4.1: Komponenten des Zerspanungsvolumens eines zylindrischen Gesenkfräasers

Für die Modellierung des Bewegungsvolumens von Grundkörpern wird zuerst die relative Bewegungskurve des Grundkörpers anhand eines Kontrollpunktes errechnet. Der Mittelpunkt der Werkzeuggrundfläche kann als ein Kontrollpunkt ausgewählt werden, der zusammen mit der Werkzeugachse die Positionen und die Orientierungen des Werkzeugs innerhalb eines Bewegungsintervalls definiert. Die kritischen Kanten auf dem Grundkörper bilden ein sogenanntes Sweeping-Pattern, welches sich entlang der Bewegungskurve bewegt. Das daraus resultierende Volumen kann durch einen Sweeping-Algorithmus nachgebildet werden.

Ausgehend von der Bereitstellung der durch die Oberflächen dargestellten Grundkörper und deren Bewegungsvolumen sind Voxelvolumen in der Repräsentationsform des Stixelmodells zur Vorbereitung der Volumenkombinationen zu erzeugen. Da je nach Grundkörpern und deren Bewegungsform innerhalb des Bewegungsintervalls das Volumen sehr unterschiedliche Formen annehmen kann, ist ein robustes Verfahren einzusetzen, welches weitergehend unabhängig von der Form eine Diskretisierung erzielt. Als vorteilhaft hat sich eine gewöhnliche Strahlenmethode erwiesen, die anhand eines Anzahlkriteriums die Schnittpunkte am Eintritt und am Austritt der Oberflächen des Grundkörpers oder deren Bewegungsvolumens ermittelt [24]. Nach der Modellierung von unterschiedlichen Voxelmodellen erfolgen zur Gestaltung des Zerspanungsvolumens die Booleschen Operationen zwischen den Voxelmodellen. Schließlich wird das Zerspanungsvolumen mit dem voxelbasierten Materialmodell zur Materialabtragung Boolesch kombiniert.

4.1.2 Klassifizierung der Grundgeometrie zur Darstellung von diversen Fräsern

Eine einheitliche Lösung für Gestaltung des Zerspanungsvolumens der verschiedenen Fräserarten für die fünfachsige NC-Simulation ist nicht möglich. Die verschiedenen Fräserarten zwingen zu einer differenzierten Betrachtung der sich daraus ergebenden Fallunterscheidungen. Jedoch kann in der voxelbasierten fünfachsigen NC-Simulation das Zerspanungsvolumen durch die Bewegungsvolumen der diversen Grundkörper wie Zylinder, Kugel, Torus, Fass und Kegel zusammengesetzt werden. Eine Voraussetzung hierfür sind die Klassifizierung dieser Grundkörper aus unterschiedlichen Fräserarten und die Generierung entsprechender Bewegungsvolumen. Bild 4.2 stellt eine allgemeine Definition der Fräsergeometrie nach DIN 66215 [88] dar.

Wie Bild 4.2 zeigt, wird die Geometrie des Werkzeugs durch sieben Parameter allgemein definiert: Werkzeugdurchmesser, Abrundungsradius, horizontaler Abstand, vertikaler Abstand, Winkel zwi-

schen Horizontale und Stirnseite, Winkel zwischen Vertikale und Mantellinie sowie Werkzeuglänge. Aus der Eingabe dieser Parameter ergeben sich dann unterschiedliche Werkzeugformen (Bild 4.3).

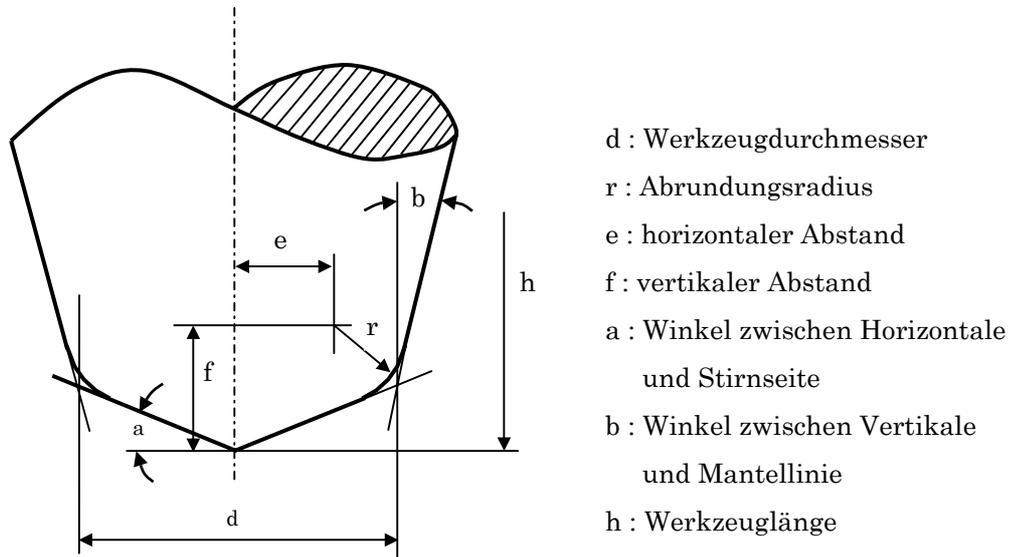


Bild 4.2: Allgemeine Definition der Fräsergeometrie nach DIN 66215

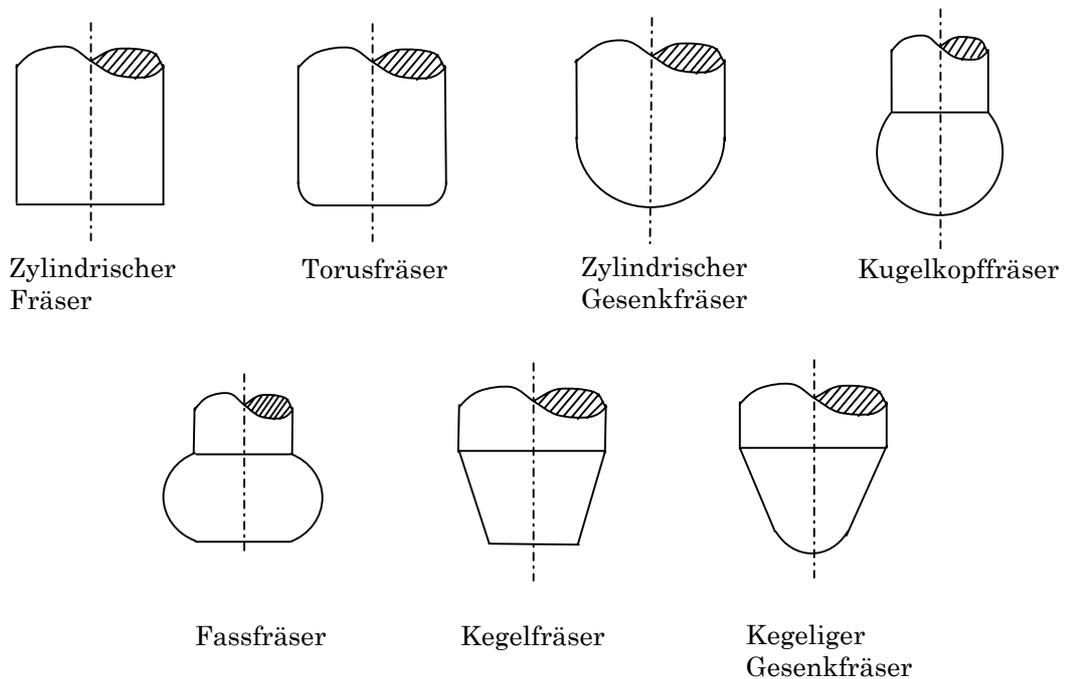


Bild 4.3: Verwendete Fräserarten für das fünffache Fräsen nach DIN 66125

In Bild 4.3 ist zu ersehen, dass die Grundkörper zur Zusammensetzung der unterschiedlichen Fräsertypen in folgenden Grundprimitiven klassifiziert werden: Zylinder, Kugel, Torus, Kegel und Fass. Diese Klassifizierung dient der einfachen Gestaltung des Bewegungsvolumens des Werkzeugs in der voxelbasierten NC-Simulation.

4.1.3 Methode zur Darstellung der Oberflächen des Grundkörpers und dessen Bewegungsmodells

Der Grundgedanke für die Modellierung von Volumen besteht darin, dass ein Körper durch seine Volumen begrenzenden Flächen definiert ist und auch so dargestellt werden kann. Wenn alle Flächen des Körpers definiert worden sind, ist er eindeutig beschrieben. Der Gedanke gilt analog für die Darstellung des Bewegungsvolumens der im Abschnitt 4.1.2 genannten Grundkörper.

Die Oberflächen der Komponenten des Zerspanungsvolumens kann man in zwei Kategorien, analytisch beschreibbare und analytisch nicht beschreibbare Flächen, unterteilen. Die analytisch beschreibbaren Flächen nach Bild 4.3 bestehen aus Zylinder, Kugel, Kegel und Fass und lassen sich durch die Geometrie, räumliche Lage und die Richtung des Werkzeugs rechnerintern exakt analytisch darstellen. Es ergeben sich folgende Vorteile:

- hohe Genauigkeit,
- kleine Datenmenge im Vergleich zu analytisch nicht beschreibbaren Modelltypen,
- keine Konsistenzprobleme,
- geringerer Aufwand zur Berechnung der Schnittpunkte zwischen den analytisch beschreibbaren Flächen und einzelner Stixel.

Die analytisch nicht beschreibbaren Flächen sind die freigeformten Flächen, die durch die kritischen Kurven auf dem bewegten Grundkörper innerhalb eines Bewegungsintervalls erzeugt werden. Zu deren Darstellung werden zwei Methoden herangezogen:

- B-Spline-Flächen und
- Dreiecksnetz.

Mit regelmäßigen, rechteckigen Parametergebieten der B-Spline-Flächen folgen Grenzfälle, in denen ein Flächenmodell nicht durch Einzelflächen geschlossen werden kann. Die klassischen Beispiele sind Lücken in lokal konkaven oder konvexen Flächenverbänden. Es hat zur Folge, dass das Volu-

menmodell nicht konsistent sein kann. Konsistenz bedeutet hier, dass keine „Löcher“ oder „Überschneidungen“ existieren. Des Weiteren ist zu beobachten, dass die Ermittlung des Schnittpunkts zwischen B-Spline-Flächen und Einzelstrahlen zeitaufwendig ist. Der Diskretisierungsaufwand des durch Flächen dargestellten Volumens zum Voxelmodell erhöht sich, wenn zahlreiche iterative Ermittlungen von Schnittpunkten durchgeführt werden müssen.

Dagegen hat das Dreiecksnetz bei der Modellierung des Volumenmodells und darauf basierten Operationen wie Berechnung des Schnittpunkts viele Vorteile. Zum ersten hat das Dreiecksnetz kein Konsistenzproblem zur Annäherung von Volumenmodellen, da sich zwei benachbarte Dreiecke durch eine einzige lineare Kante ohne Spalt verbinden lassen. Zum anderen ist die rechnerinterne Darstellung des Dreiecksnetzes im Vergleich zu den anderen rechnerinternen Repräsentationen einfacher. Bei der rechnerinternen Darstellung eines Dreiecks werden lediglich die drei Eckpunkte des Dreiecks gespeichert. Zum dritten können die Algorithmen, welche die Operationen auf dem Dreieck ermöglichen, auf einfache Weise aufgebaut werden [89]. Im Vergleich zur B-Spline-Fläche lässt sich die Ermittlung des Schnittpunkts zwischen einem Dreieck und einzelner Stixel einfach und schnell durchführen. Die Nachteile des Dreiecksnetzes zur Darstellung eines komplex geformtem Flächenmodells bestehen in dem hohen Speicherbedarf [90].

In der vorliegenden Arbeit werden Dreiecksnetze zur Darstellung der Oberflächen des Bewegungsvolumens von Werkzeuggrundkörpern eingesetzt.

4.1.4 Rechnerinterne Informationsstruktur der Fräser

Im Abschnitt 4.1.2 wurde gezeigt, dass das Zerspanungsvolumen des Werkzeugs durch die Grundkörper und deren Bewegungsvolumen zusammengesetzt werden kann. Aufgrund der Vielfalt möglicher Werkzeuge ist es erforderlich, ihre gemeinsamen Eigenschaften anhand einer Klassifizierung zu einer Methode zusammenzufassen und dann als spezifische Werkzeuge bereitzustellen. Dieser Sachverhalt wird über einen objektorientierten Ansatz umgesetzt, der über den Mechanismus der Vererbung Daten und Methoden einer Basisklasse den abgeleiteten Klassen zur Verfügung stellt (Bild 4.4).

Im Verlauf der NC-Simulation sind wiederholt die folgenden Aufgaben zur rechnerinternen Darstellung des Zerspanungsvolumens durchzuführen:

- Analytische Repräsentation von Grundkörpern an Anfangs- und Endposition eines Bewegungsintervalls,

- Berechnung des dreiecksbasierten Bewegungsvolumens des Grundkörpers innerhalb eines Bewegungsintervalls,
- Umsetzung des dreiecksbasierten Bewegungsvolumens in ein voxelbasiertes Volumenmodell und
- Boolesche Addition der diversen Voxelmodelle zur Generierung des vollständigen Zerspanungsvolumens innerhalb eines Bewegungsintervalls.

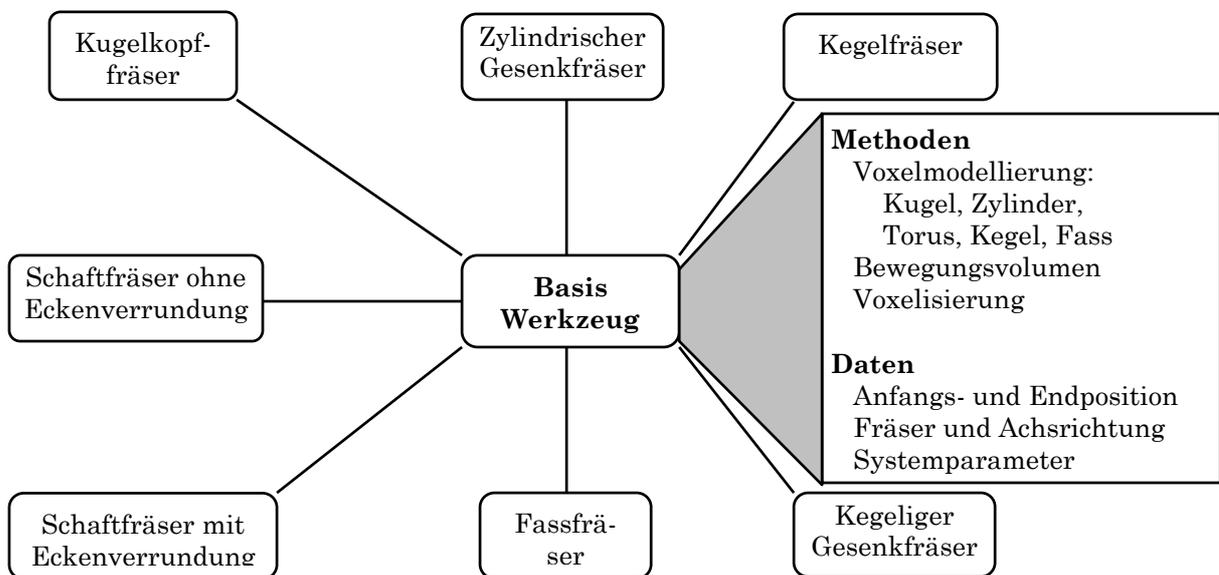


Bild 4.4: Rechnerinterne Informationsstruktur der Fräser

Da sich das Zerspanungsvolumen in der voxelbasierten NC-Simulation prinzipiell aus den Grundkörpern und deren Bewegungsvolumen zusammensetzt, sind die Grundkörper wie Kugel, Torus, Zylinder, Kegel und Fass und deren Bewegungsvolumen allgemeingültig als Basismodelle in dem Basiswerkzeug zu modellieren. Dazu werden die NC-Bahnen (Anfangs- und Endposition und entsprechende Achsrichtungen) und Werkzeugparameter gespeichert. Da die Umsetzung von Grundkörpervolumen und deren Bewegungsvolumen im Voxelmodell unabhängig von der Gestalt der Grundkörper ist, wird auch die Umsetzungsmethode als ein allgemeiner Ansatz im Basiswerkzeug platziert.

Ein abgeleitetes Werkzeug stellt Datenfelder und Methoden zur Definition und Darstellung der Geometrie des Werkzeugs bereit. Die Eingangsparameter sind die Werkzeugparameter und die CLDA-

CLDATA. Dagegen wird in der abgeleiteten Werkzeugklasse je nach Eigenschaften des Werkzeugs die Methode zur Booleschen Addition von den in der Basisklasse erzeugten Voxelmolellen bereitgestellt.

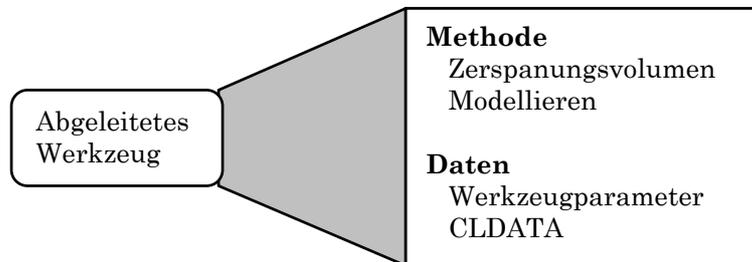


Bild 4.5: Abgeleitetes Werkzeug

Zur graphischen Darstellung und Booleschen Operationen von Zerspanungsvolumen mit Materialmodell werden die Methoden, die im VCM-System entwickelt sind, direkt eingesetzt.

4.2 Gestaltung des Bewegungsvolumens der diversen Grundkörper in der fünfsichtigen NC-Simulation

4.2.1 Gestaltung des Bewegungsvolumens einer Kugel

Bild 4.6 stellt eine sich auf einer NC-Bahn bewegendende Kugel dar. Die bewegte Kugel lässt sich in der folgenden Formel beschreiben:

$$\vec{r}_{(\theta, \phi, t)} = \vec{R}_{(t)} + a \cos \theta \vec{e}_1 + a \sin \theta \cos \phi \vec{e}_2 + a \sin \theta \sin \phi \vec{e}_3 \quad (4-1)$$

wobei

$\vec{R}_{(t)}$: NC-Bahnpunkt in der Zeit t , $0 \leq t \leq 1$

a : Kugelradius

θ : $0 \leq \theta \leq 2\pi$

ϕ : $0 \leq \phi \leq 2\pi$

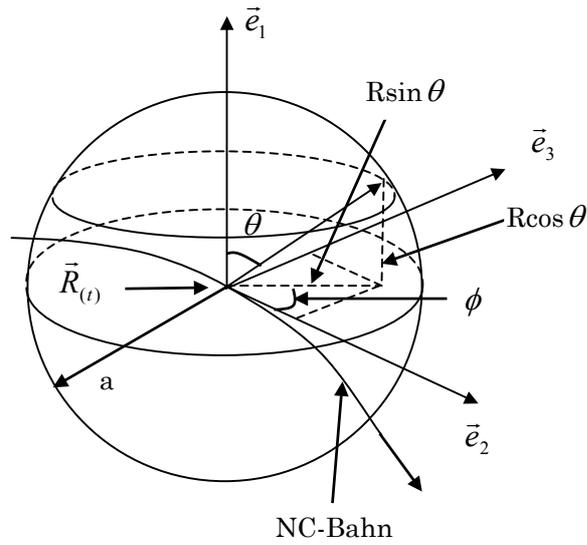


Bild 4.6: Eine Kugel bewegt sich auf einer NC-Bahn

Die Basis des bewegten Frames $\vec{e}_1, \vec{e}_2, \vec{e}_3$ wird wie folgt definiert:

$$\begin{aligned}\vec{e}_1 &= \vec{A} \\ \vec{e}_2 &= \frac{\dot{\vec{A}}}{|\dot{\vec{A}}|} \\ \vec{e}_3 &= \vec{e}_1 \times \vec{e}_2\end{aligned}$$

\vec{A} ist der Einheitsvektor der Werkzeugachse und wird durch winklig lineare Interpolation von den Werkzeugachsen \vec{A}_1 und \vec{A}_2 jeweils an Anfangs- und Endposition des Bewegungsintervalls berechnet. Somit ergibt sich folgende Beziehung:

$$\vec{A} = (\cos(\theta_p * t) \frac{\vec{A}_1}{\sqrt{\vec{A}_1 \bullet \vec{A}_2}} + \sin(\theta_p * t) \frac{\vec{P}}{\sqrt{\vec{P} \bullet \vec{P}}}) \sqrt{\vec{A}_1 \bullet \vec{A}_2} \quad (4-2)$$

wobei

$$t: 0 \leq t \leq 1$$

$$\theta_p = \cos^{-1}(\vec{A}_1 \bullet \vec{A}_2)$$

$$\vec{P} = (\vec{A}_1 \times \vec{A}_2) \times \vec{A}_1$$

Die Oberflächennormale des Werkzeugs \vec{N} und die Geschwindigkeit des Oberflächenpunkts \vec{V} lassen sich wie folgt ableiten:

$$\vec{N} = \dot{\vec{r}}_\theta \times \dot{\vec{r}}_\phi = a^2 \sin \theta (\cos \theta \vec{e}_1 + \sin \theta \cos \phi \vec{e}_2 + \sin \theta \sin \phi \vec{e}_3) \quad (4-3)$$

$$\vec{V} = \dot{\vec{r}}_{(t)} = \dot{\vec{R}}_{(t)} + a \cos \theta \left| \dot{\vec{A}} \right| \vec{e}_2 - a \sin \theta \cos \phi \left| \dot{\vec{A}} \right| \vec{e}_1 \quad (4-4)$$

Werden die Formeln (4-3), (4-4) in Formel (2-1) eingesetzt, ergibt sich die folgende Beziehung:

$$\theta = \operatorname{tg}^{-1} \frac{-\dot{\vec{R}}_{(t)} \cdot \vec{e}_1}{\cos \phi \dot{\vec{R}}_{(t)} \cdot \vec{e}_2 + \sin \phi \dot{\vec{R}}_{(t)} \cdot \vec{e}_3} \quad (4-5)$$

Es ist offensichtlich, dass eine Kugel innerhalb des Bewegungsintervalls ein zylindrisches Volumen durchfährt. Bild 4.7 zeigt die erzeugten kritischen Kurven auf der Kugeloberfläche.

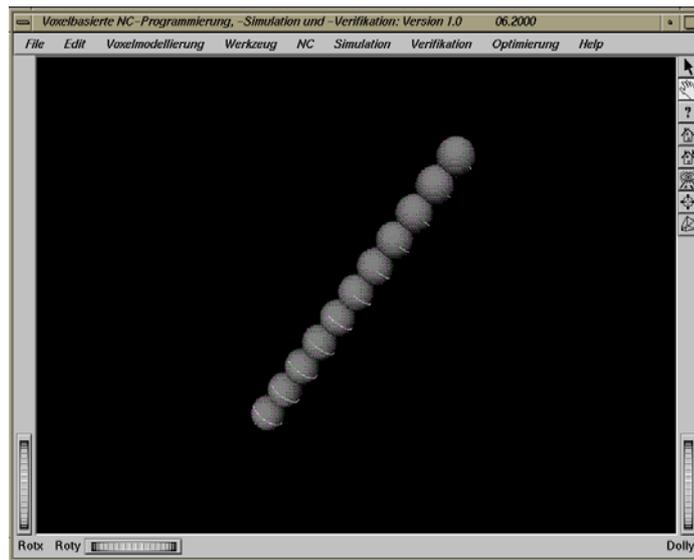


Bild 4.7: Erzeugte kritische Kurven auf der Kugelfläche

4.2.2 Gestaltung des Bewegungsvolumens eines Zylinders

Bild 4.8 zeigt einen sich auf einer NC-Bahn bewegendem Zylinder. Der bewegte Zylinder lässt sich mit folgender Formel beschreiben [40].

$$\vec{r}_{(\theta, u, t)} = \vec{R}_{(t)} + u \vec{A}_{(t)} + a \cos \theta \vec{e}_2 + a \sin \theta \vec{e}_3 \quad (4-6)$$

wobei

$\vec{R}_{(t)}$: NC-Bahnpunkt in der Zeit t , $0 \leq t \leq 1$

l : Werkzeuglänge

u : $0 \leq u \leq l$

a : Zylinderradius

θ : $0 \leq \theta \leq 2\pi$

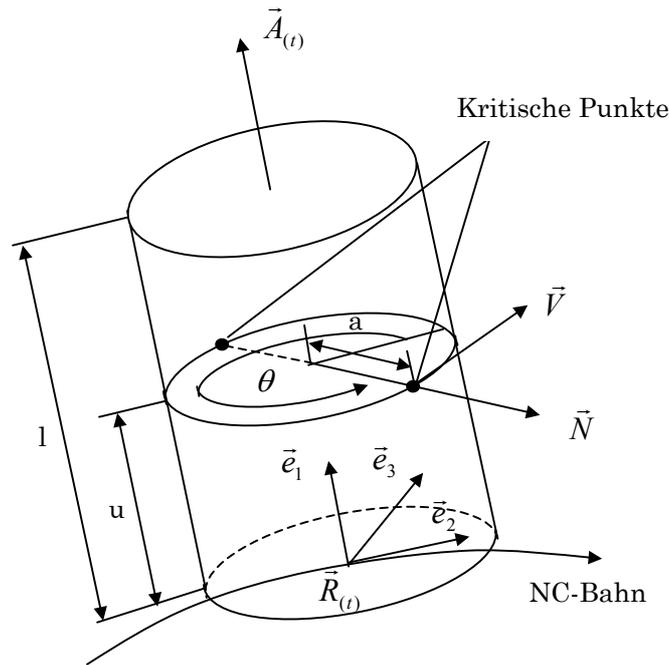


Bild 4.8: Ein Zylinder bewegt sich auf einer NC-Bahn

Die Basis des bewegten Frames $\vec{e}_1, \vec{e}_2, \vec{e}_3$ wird wie im obigen Abschnitt 4.2.1 definiert.

Die Oberflächennormale des Zylinders \vec{N} und die Geschwindigkeit des Oberflächenpunkts \vec{V} lassen sich wie folgt ableiten:

$$\vec{N} = \dot{\vec{r}}_\theta \times \dot{\vec{r}}_u = \cos \theta \vec{e}_2 + \sin \theta \vec{e}_3 \quad (4-7)$$

$$\vec{V} = \dot{\vec{r}}_{(t)} = \dot{\vec{R}}_{(t)} + u \left| \dot{\vec{A}} \right| \vec{e}_2 - a \cos \theta \left| \dot{\vec{A}} \right| \vec{e}_1 \quad (4-8)$$

Nach Formel (2-1) ergibt sich die folgende Beziehung:

$$\theta_{(u,t)} = \tan^{-1} \left(- \frac{\vec{e}_2 \bullet \vec{R}_{(t)} + u \left| \dot{\vec{A}} \right|}{\vec{e}_3 \bullet \vec{R}_{(t)}} \right) \quad (4-9)$$

Es ist festzustellen, dass $\theta_{(u,t)}$ eine stetige Funktion von u zwischen 0 und 1 ist und im Bereich $0 \leq \theta \leq 2\pi$ zwei Lösungen vorliegen. Somit sind zwei kritische Kurven der Zylinderfläche abzuleiten. Falls sich der Zylinder lediglich innerhalb des Bewegungsintervalls parallel bewegt, nämlich $\dot{\vec{A}} = 0$ und $\vec{A} \times \dot{\vec{R}} = 0$, dann ergibt sich $\theta = \pi/2$ und $\theta = 3\pi/2$. Dies bedeutet, dass die Envelopfläche des Bewegungsvolumens eines Zylinders in diesem Fall direkt durch Bewegung der Außenkontur des Zylinders, beobachtet in der Richtung $\dot{\vec{R}}$, erzeugt werden kann. Bild 4.9 zeigt die erzeugten kritischen Kurven der Zylinderoberfläche.

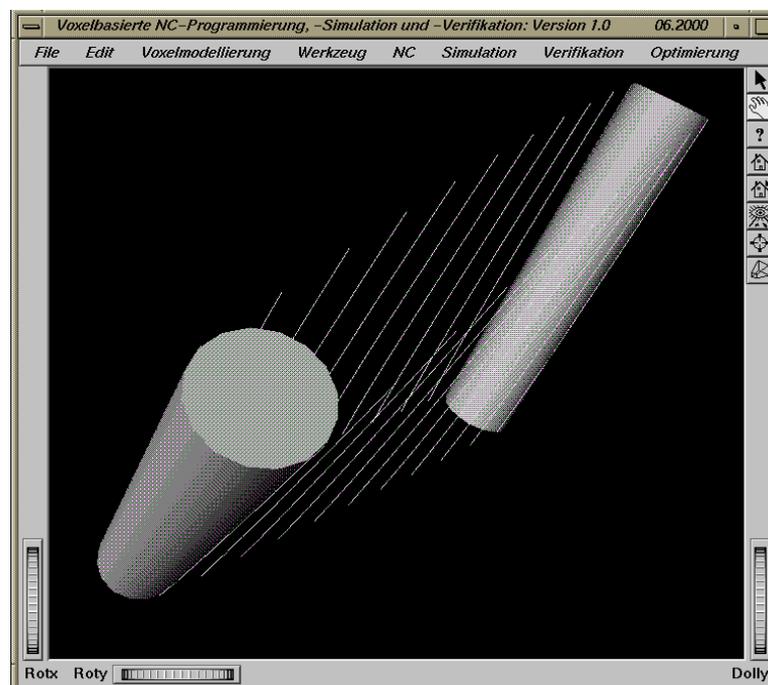


Bild 4.9: Erzeugte kritische Kurven der Zylinderoberfläche

4.2.3 Gestaltung des Bewegungsvolumens eines Torus

In Bild 4.10 wird ein sich auf einer NC-Bahn bewegendes Torus gezeigt. Der bewegte Torus lässt sich mit folgender Formel beschreiben.

$$r_{(\alpha,\theta,t)} = \vec{R}_{(t)} + a \sin \theta \vec{e}_1 + (b + a \cos \theta) \cos \alpha \vec{e}_2 + (b + a \cos \theta) \sin \alpha \vec{e}_3 \quad (4-10)$$

wobei

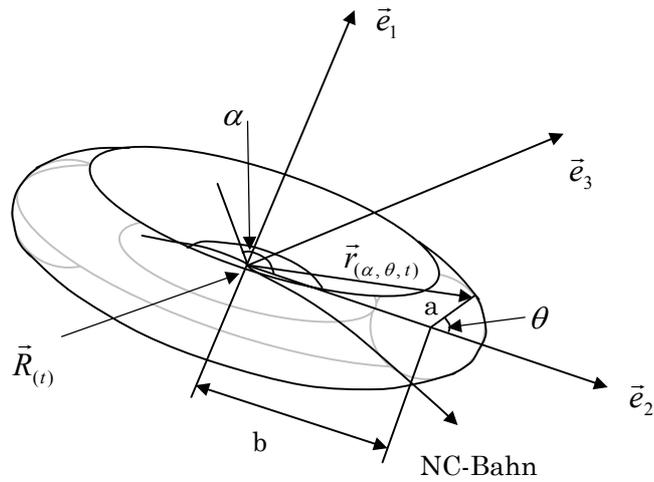


Bild 4.10: Ein Torus bewegt sich auf einer NC-Bahn

$\vec{R}_{(t)}$: NC-Bahnpunkt in der Zeit t , $0 \leq t \leq 1$

a : zweiter Radius des Torus

b : erster Radius des Torus

α : $0 \leq \alpha \leq 2\pi$

θ : $0 \leq \theta \leq 2\pi$

Die Basis des bewegten Frames $\vec{e}_1, \vec{e}_2, \vec{e}_3$ wird wie im obigen Abschnitt 4.2.1 definiert.

Die Oberflächennormale des Torus \vec{N} und die Geschwindigkeit des Oberflächenpunkts \vec{V} lassen sich wie folgt ableiten:

$$\vec{N} = \dot{\vec{r}}_\alpha \times \dot{\vec{r}}_\theta = -a(b + a \cos \theta)(\sin \theta \vec{e}_1 + \cos \theta \cos \alpha \vec{e}_2 + \cos \theta \sin \alpha \vec{e}_3) \quad (4-11)$$

$$\vec{V} = \dot{\vec{r}}_t = \dot{\vec{R}}_{(t)} + a \sin \theta \left| \dot{\vec{A}} \right| \vec{e}_2 - (b + a \cos \theta) \cos \alpha \left| \dot{\vec{A}} \right| \vec{e}_1 \quad (4-12)$$

Nach Formel (2-1) ergibt sich die folgende Beziehung:

$$\theta = \text{tg}^{-1} \left(\frac{\cos \alpha \vec{e}_2 \cdot \dot{\vec{R}} + \sin \alpha \vec{e}_3 \cdot \dot{\vec{R}}}{r \cos \alpha \left| \dot{\vec{A}} \right| - \vec{e}_1 \cdot \dot{\vec{R}}} \right) \quad (4-13)$$

Es ist zu erkennen, dass $\theta_{(u,t)}$ eine stetige Funktion mit α zwischen 0 und 2π ist und im Bereich $0 \leq \theta \leq 2\pi$ zwei Lösungen hat. Somit sind zwei kritische Kurven auf der Torusfläche abzuleiten. Für die äußere Fläche des Bewegungsvolumens des Torus gilt θ im Bereich $-\pi/2 < \theta < \pi/2$. Für die innere Fläche des Bewegungsvolumens des Torus gilt θ im Bereich $\pi/2 < \theta < 3\pi/2$.

Bild 4.11 stellt die ermittelten kritischen Kurven auf der äußeren und der inneren Oberfläche des bewegten Torus dar.

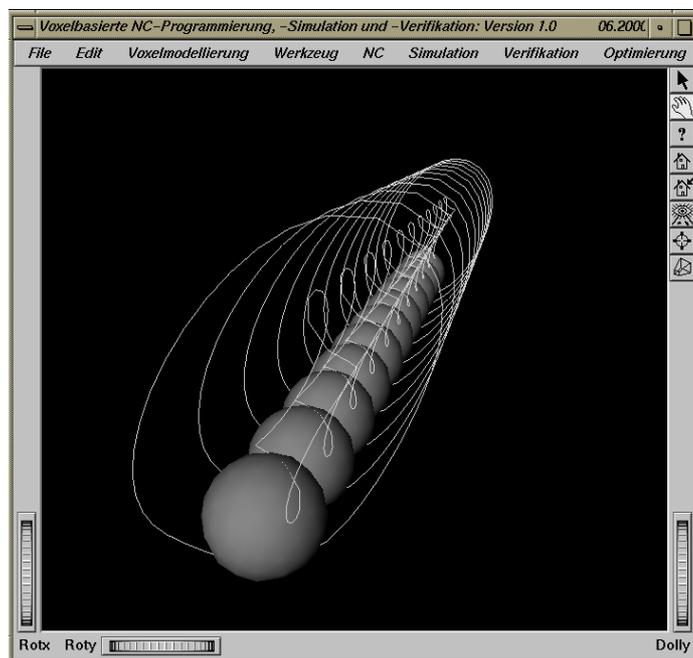


Bild 4.11: Kritische Kurven auf der äußeren und inneren Oberfläche des bewegten Torus

Bild 4.12 zeigt die ermittelten kritischen Kurven auf der Oberfläche eines Zylinders und der Oberfläche eines Torus. Für den Torus werden hier nur die äußeren kritischen Kurven auf der Oberfläche des Bewegungsvolumens des Torus dargestellt, da für die Gestaltung des Bewegungsvolumens eines Torusfräsers nur die äußeren kritischen Kurven des Torus beteiligt sind.

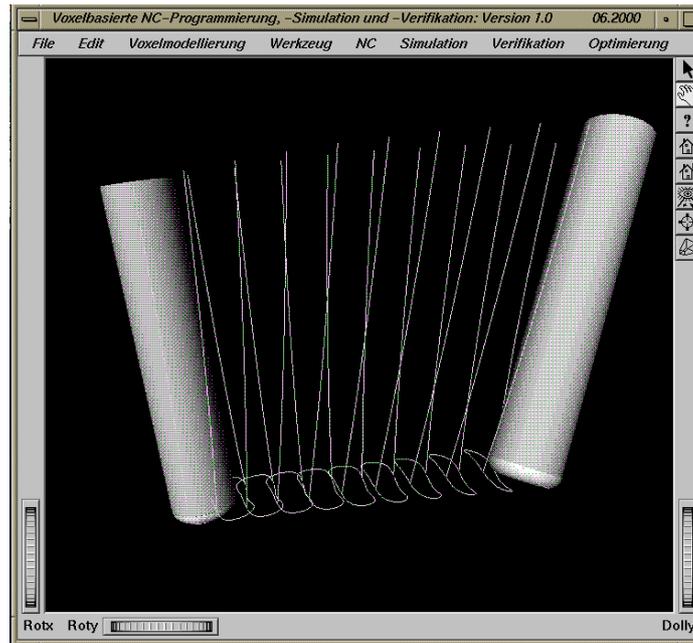


Bild 4.12: Die ermittelten kritischen Kurven auf der Oberfläche eines Zylinders und der Oberfläche eines Torus

4.2.4 Gestaltung des Bewegungsvolumens eines Kegels

Bild 4.13 zeigt einen sich auf einer NC-Bahn bewegenden Kegel. Der bewegte Kegel wird mit folgender Formel beschrieben.

$$\vec{r}_{(\theta, u, t)} = \dot{\vec{R}}_{(t)} + u \left| \vec{A} \right| \vec{e}_1 + b_{(u)} \cos \theta \vec{e}_2 + b_{(u)} \sin \theta \vec{e}_3 \quad (4-14)$$

wobei

$\vec{R}_{(t)}$: NC-Bahnpunkt in der Zeit t , $0 \leq t \leq 1$

u : $0 \leq u \leq l$

θ : Schrägwinkel

a : Radius der Grundfläche des Kegels

$$b_{(u)} = a + u \left| \vec{A} \right| \operatorname{tg} \phi$$

Die Basis des bewegten Frames $\vec{e}_1, \vec{e}_2, \vec{e}_3$ wird wie im obigen Abschnitt 4.2.1 definiert.

Die Oberflächennormale des Kegels \vec{N} und die Geschwindigkeit des Oberflächenpunkts \vec{V} lassen sich wie folgt ableiten:

$$\vec{N} = \dot{\vec{r}}_u \times \dot{\vec{r}}_\theta = \dot{b}_{(u)} b_{(u)} \vec{e}_1 - b_{(u)} \cos \theta \left| \vec{A} \right| \vec{e}_2 - b_{(u)} \sin \theta \left| \vec{A} \right| \vec{e}_3 \quad (4-15)$$

$$\vec{V} = \dot{\vec{r}}_t = \dot{\vec{R}}_{(t)} + u \left| \vec{A} \right| \left| \dot{\vec{B}} \right| \vec{e}_2 - b_{(u)} \cos \theta \left| \dot{\vec{B}} \right| \vec{e}_1 \quad (4-16)$$

Nach Formel (2-1) ergibt sich die folgende Beziehung:

$$\begin{aligned} & (-\dot{b}_{(u)} b_{(u)} \left| \dot{\vec{B}} \right| - b_{(u)} \left| \vec{A} \right| \dot{\vec{R}} \bullet \vec{e}_2 - b_{(u)} u \left| \vec{A} \right|^2 \left| \dot{\vec{B}} \right|) \cos \theta - b_{(u)} \left| \vec{A} \right| \dot{\vec{R}} \bullet \vec{e}_3 \sin \theta \\ & + \dot{b}_{(u)} b_{(u)} \dot{\vec{R}} \bullet \vec{e}_1 = 0 \end{aligned} \quad (4-17)$$

wobei $\vec{B} = \frac{\vec{A}}{\left| \vec{A} \right|}$

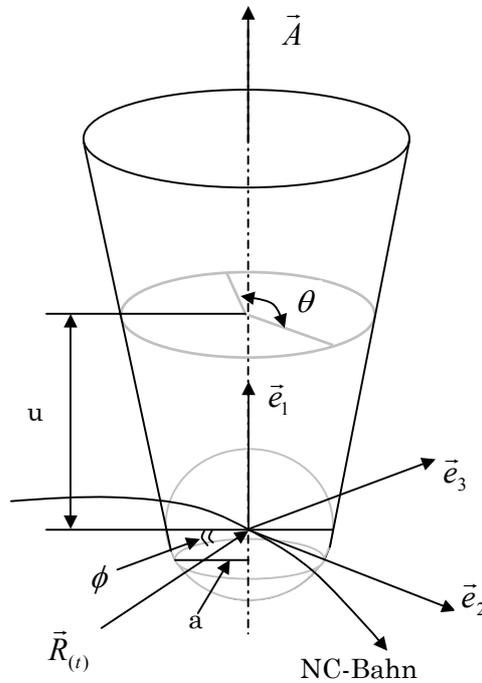


Bild 4.13: Ein Kegel und eine Kugel bewegen sich auf einer NC-Bahn

Es wird angenommen:

$$D = (-\dot{b}_{(u)} b_{(u)} \left| \dot{\vec{B}} \right| - b_{(u)} \left| \vec{A} \right| \dot{\vec{R}} \bullet \vec{e}_2 - b_{(u)} u \left| \vec{A} \right|^2 \left| \dot{\vec{B}} \right|)$$

$$E = -b_{(u)} \left| \vec{A} \right| \dot{\vec{R}} \bullet \vec{e}_3$$

$$F = \dot{b}_{(u)} b_{(u)} \dot{\vec{R}} \bullet \vec{e}_1$$

Die Formel (4-17) wird vereinfacht zu:

$$D \cos \theta + E \sin \theta + F = 0 \quad (4-18)$$

Daraus ergibt sich folgende Lösung:

$$\theta = \operatorname{tg}^{-1} \left(\frac{(D - F)(E - L)}{F^2 - DF - E^2 + EL} \right) \quad (4-19)$$

wobei

$$L = \sqrt{E^2 + D^2 - F^2} \text{ ist.}$$

Es ist zu bemerken, dass $\theta_{(u,t)}$ eine stetige Funktion mit u zwischen 0 und 1 ist und im Bereich $0 \leq \alpha \leq 2\pi$ zwei Lösungen hat. Somit sind zwei kritische Kurven auf der Kegelfläche abzuleiten.

Bild 4.14 stellt die ermittelten kritischen Kurven auf der Oberfläche des bewegten abgerundeten Kegels dar.

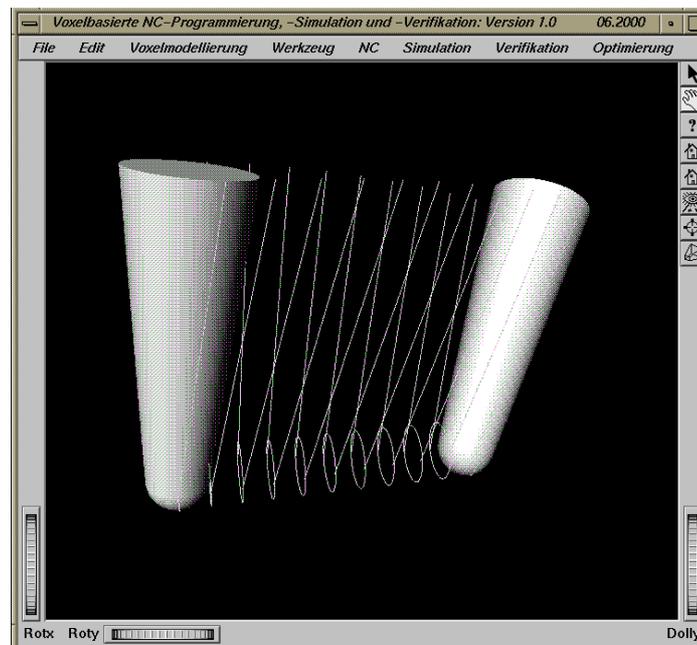


Bild 4.14: Kritische Kurven auf der Oberfläche eines Kegels und einer Kugel

4.2.5 Gestaltung des Bewegungsvolumens eines Fasses

Bild 4.15 zeigt ein sich auf einer NC-Bahn bewegendes Fass. Das bewegte Fass lässt sich mit folgender Formel beschreiben.

$$\vec{r}_{(\theta, u, t)} = \dot{\vec{R}}_{(t)} + 2H u \vec{e}_1 + b_{(u)} \cos \theta \vec{e}_2 + b_{(u)} \sin \theta \vec{e}_3 \quad (4-20)$$

wobei

$\vec{R}_{(t)}$: NC-Bahnpunkt in der Zeit t , $0 \leq t \leq 1$

u : $0 \leq u \leq 2H$

H : Halbe Länge des Fasses

θ : $0 \leq \theta \leq 2\pi$

$$b_{(u)} = a \sqrt{1 - \frac{H^2 (1 - 2u)^2}{a^2}} - \delta$$

a : Radius der Mantelfläche des Fasses

δ : Abstand des Zentrums der Fassfläche zur Fassachse

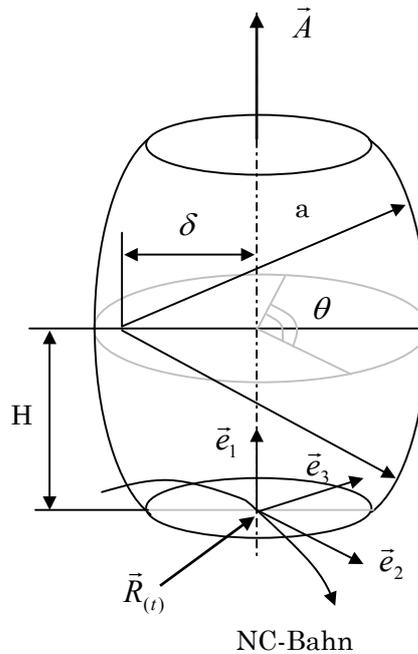


Bild 4.15: Ein Fass bewegt sich auf einer NC-Bahn

Die Basis des bewegten Frames $\vec{e}_1, \vec{e}_2, \vec{e}_3$ wird wie im obigen Abschnitt 4.2.1 definiert.

Die Oberflächennormale des Fasses \vec{N} und die Geschwindigkeit des Oberflächenpunkts \vec{V} lassen sich wie folgt ableiten:

$$\vec{N} = \dot{\vec{r}}_u \times \dot{\vec{r}}_\theta = \dot{b}_{(u)} b_{(u)} \vec{e}_1 - 2 H b_{(u)} \cos \theta \vec{e}_2 - 2 H b_{(u)} \sin \theta \vec{e}_3 \quad (4-21)$$

$$\vec{V} = \dot{\vec{r}}_t = \dot{\vec{R}}_{(t)} + 2 H u \left| \dot{\vec{B}} \right| \vec{e}_2 - b_{(u)} \cos \theta \left| \dot{\vec{B}} \right| \vec{e}_1 \quad (4-22)$$

Nach Formel (2-1) ergibt sich die folgende Beziehung:

$$\begin{aligned} & (-\dot{b}_{(u)} b^2_{(u)} \left| \dot{\vec{B}} \right| - 2Hb_{(u)} \dot{\vec{R}}_{(t)} \bullet \vec{e}_2 - 4H^2 u b_{(u)} \left| \dot{\vec{B}} \right|) \cos \theta - 2Hb_{(u)} \dot{\vec{R}} \bullet \vec{e}_3 \sin \theta \\ & + \dot{b}_{(u)} b_{(u)} \dot{\vec{R}} \bullet \vec{e}_1 = 0 \end{aligned} \quad (4-23)$$

wobei $\vec{B} = \frac{\vec{A}}{|\vec{A}|}$

Es wird angenommen:

$$\begin{aligned} D &= (-\dot{b}_{(u)} b^2_{(u)} \left| \dot{\vec{B}} \right| - 2Hb_{(u)} \left| \vec{A} \right| \dot{\vec{R}} \bullet \vec{e}_2 - 4H^2 b_{(u)} u \left| \dot{\vec{B}} \right|) \\ E &= -2Hb_{(u)} \dot{\vec{R}} \bullet \vec{e}_3 \\ F &= \dot{b}_{(u)} b_{(u)} \dot{\vec{R}} \bullet \vec{e}_1 \end{aligned}$$

Die Formel (4-23) wird vereinfacht zu:

$$D \cos \theta + E \sin \theta + F = 0 \quad (4-24)$$

Daraus ergibt sich folgende Lösung:

$$\theta = \operatorname{tg}^{-1} \left(\frac{(D - F)(E - L)}{F^2 - DF - E^2 + EL} \right) \quad (4-25)$$

wobei

$$L = \sqrt{E^2 + D^2 - F^2} \text{ ist.}$$

Es ist festzustellen, dass $\theta_{(u,t)}$ eine stetige Funktion mit u zwischen 0 und $2H$ ist und im Bereich $0 \leq \alpha \leq 2\pi$ zwei Lösungen hat. Somit sind zwei kritische Kurven auf der Fassfläche abzuleiten.

Bild 4.16 stellt die ermittelten kritischen Kurven auf der Oberfläche des bewegten Fasses dar.

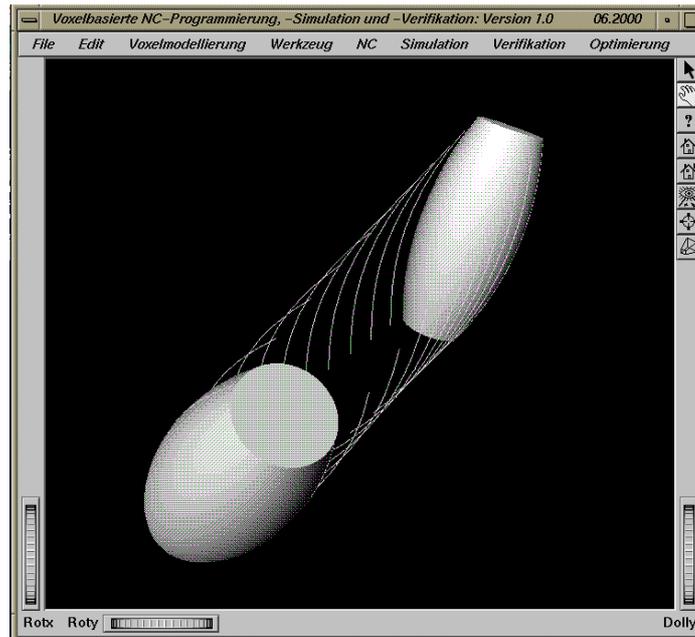


Bild 4.16: Kritische Kurven auf der Fläche eines bewegten Fasses

4.2.6 Gestaltung der Oberflächen der diversen Bewegungsvolumen mit Dreiecksnetzen

Zur Generierung eines Voxelmodells aus Bewegungsvolumen der im Abschnitt 4.1 definierten Grundkörper müssen die Oberflächen des Bewegungsvolumens durch kritische Kurven erzeugt werden. Nach dem Konzept im Abschnitt 4.1 sollen diese Oberflächen des Bewegungsvolumens durch Dreiecksnetze angenähert werden. Bild 4.17 zeigt ein Beispiel der Oberflächen des Bewegungsvolumens eines Zylinders. Nachfolgend werden drei Schritte zur Gestaltung der dreiecksbasierten Oberflächen des Bewegungsvolumens für Zylinder, Kegel und Fass dargestellt:

- Generierung der Seitenflächen F_1 und F_2 über ihre kritischen Kurven, die innerhalb des Zeitintervalls $[0, 1]$ durch die im Abschnitt 4.2.5 dargestellten Algorithmen berechnet werden,
- Generierung der Anfangs- und Endflächen F_3 und F_4 über jeweils zwei kritische Kurven zum Zeitpunkt 0 und 1 und
- Generierung der Top- und Grundflächen F_5 und F_6 über die Anfangs- und Endpunkte der kritischen Kurven.

Es ist offensichtlich, dass das Bewegungsvolumen einer Kugel innerhalb eines Bewegungsintervalls eigentlich ein Zylinder ist. Dieser Zylinder kann direkt durch eine analytisch beschreibbare Fläche dargestellt werden. Um die einheitliche Darstellung der Bewegungsvolumen und die Vorteile der

objektorientierten Programmierung einzuhalten, wird das Bewegungsvolumen der Kugel innerhalb eines Bewegungsintervalls durch Dreiecksnetze angenähert.

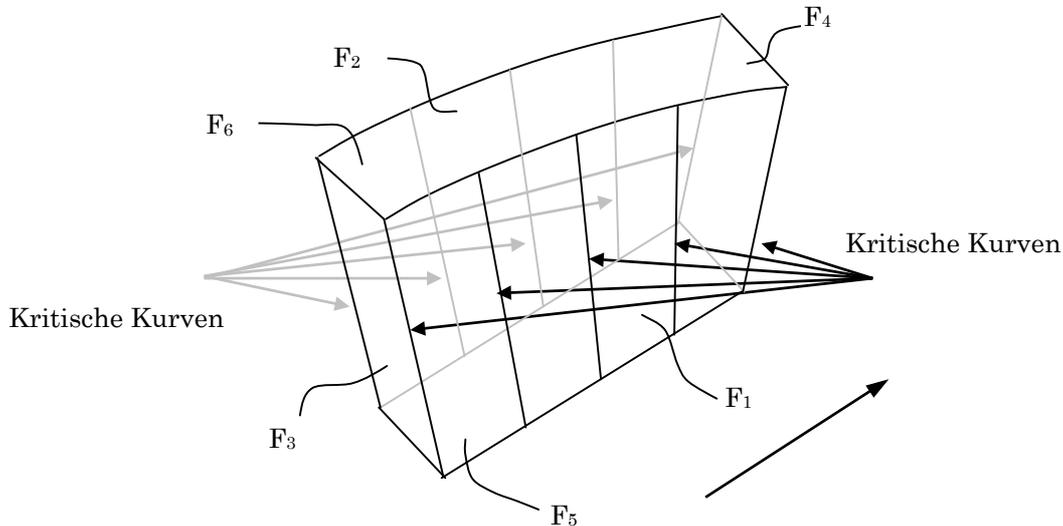


Bild 4.17: Beispiel: Gestaltung der Oberflächen des Bewegungsvolumens

Bei der Generierung des Bewegungsvolumens eines Torus innerhalb eines Bewegungsintervalls ist zu beachten, dass die kritischen Kurven des Torus in zwei Teile unterteilt werden, nämlich die inneren kritischen Kurven und die äußeren kritischen Kurven. Die inneren kritischen Kurven befinden sich innerhalb des Bereichs $\pi/2 \leq \theta \leq 3\pi/2$ des bewegten Torus (Bild 4.11) und sind bei der Gestaltung des Bewegungsvolumens nicht beteiligt. Die äußeren kritischen Kurven hingegen bilden das dreiecksbasierte Bewegungsvolumen des Torus innerhalb eines Bewegungsintervalls und sind innerhalb des Bereichs $-\pi/2 \leq \theta \leq \pi/2$ zu ermitteln.

Die Genauigkeit des Bewegungsvolumens hängt direkt von der Anzahl der kritischen Kurven und der Anzahl der Punkte auf jeder kritischen Kurve ab. Je größer die Anzahl beider ist, desto größer die Anzahl der Dreiecke und um so höher die Genauigkeit des Bewegungsvolumens. Der mit Anzahl der Dreiecke und nachgeschalteter Ermittlung der Schnittpunkte verbundene Rechenaufwand setzt jedoch Grenzen für die Anzahl der Punkte sowie für die Anzahl der kritischen Kurven. In der vorliegenden Arbeit wird das Bewegungsintervall $t[0, 1]$ in 4 gleiche Teile gegliedert und jede kritische Kurve mit 5 Punkten berechnet.

4.3 Voxelbasierte Diskretisierung der Grundkörper und Bewegungsvolumen

4.3.1 Diskretisierung des Grundkörpers

Gemäß Abschnitt 4.1 werden die Grundkörper zur Zusammensetzung der unterschiedlichen Fräser-typen in Kugel, Zylinder, Torus, Kegel und Fass klassifiziert, die durch analytisch beschreibbare Flächen dargestellt werden können. Die analytisch beschreibbare Flächen setzen sich aus Kugelfläche, Torusfläche und Mantelflächen zusammen. Die Mantelflächen sind Zylinder-, Kegel-, und Fassmantelflächen. Die beiden Stirnflächen des beschriebenen Zylinders, Kegels oder Fasses sind Ebenen, womit Zylinder, Kegel oder Fass geschlossen werden können.

Die Diskretisierung einer Kugel erfolgt durch die direkte Berechnung der Schnittpunkte zwischen analytisch dargestellter Kugel und achsparallelen Strahlen (siehe Stixelmodell). Falls keiner oder lediglich ein Schnittpunkt ermittelt wird, wird kein Stixelmodell für die Kugel erzeugt. Zur Diskretisierung des Torus müssen zuerst zwei Kreisebenen wie Grund- und Deckkreisebenen mit dem Radius ($r_f - r_e$) zum Schließen des Torus definiert werden. Die Berechnungen der Schnittpunkte werden zwischen achsparallelen Strahlen und analytisch dargestelltem Torus sowie zwei Kreisebenen durchgeführt. Falls mehr als 2 Schnittpunkte ermittelt werden, muss zuerst die Reihenfolge der Schnittpunkte genau ermittelt werden. Danach kann ein Stixel für den Torus bestimmt werden. Liegen die Strahlen auf der Kreisebene, so lässt sich ein Stixel mit der Länge $\eta(2r_f - 2r_e)$ für den Torus bilden. Hier gilt $0 \leq \eta \leq 1.0$. Falls kein oder lediglich ein Schnittpunkt berechnet wird, so wird kein Stixelmodell für den Torus gebildet.

Zur Diskretisierung von Zylinder, Kegel und Fass müssen zuerst zwei Kreisebenen wie Grundkreisebene und Deckkreisebene zum Schließen der entsprechenden Volumen erzeugt werden. Die Berechnungen der Schnittpunkte lassen sich zwischen achsparallelen Strahlen und Mantelfläche sowie zwei Kreisebenen ausführen. Falls zwei Schnittpunkte berechnet werden, lässt sich ein Stixel bestimmen. Wird kein oder lediglich ein Schnittpunkt ermittelt, gibt es in diesem Fall kein Stixel.

4.3.2 Diskretisierung der dreiecksbasierten Bewegungsvolumen

Die Oberflächen des Bewegungsvolumens eines Grundkörpers sind freigeformte Flächen, die in der vorliegenden Arbeit durch Dreiecksnetze angenähert werden. Für die dreiecksbasierten Oberflächen sind vier Berechnungsschritte zur Ermittlung des Schnittpunkts durchzuführen:

- Gestaltung eines Dreieckes,

- Beschränkung der Anzahl der Strahlen durch einen Hüllwürfel,
- Schnittpunktermittlung und
- Ermittlung der Existenz eines Schnittpunkts in diesem Dreieck.

4.3.2.1 Gestaltung eines Dreieckes

Ein Dreieck in dem Dreiecksnetz wird mittels folgender Methode im System definiert. Es wird festgelegt, aus welchen der auf den kritischen Kurven befindlichen Punkten das Dreieck besteht. Dadurch wird bei der Berechnung des Schnittpunkts im ersten Schritt ein Dreieck gebildet. Bild 4.18 zeigt beispielsweise ein entsprechendes Dreieck $P_1P_2P_3$.

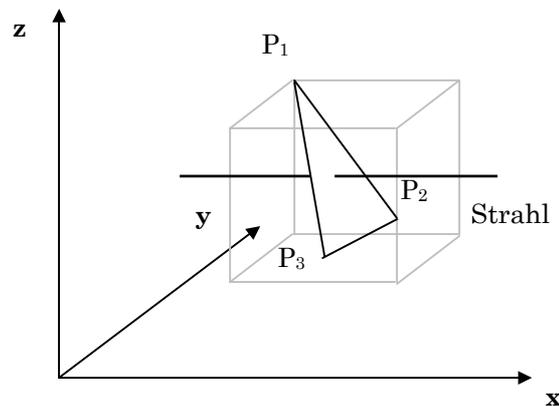


Bild 4.18: Ein Dreieck und ein Hüllwürfel

4.3.2.2 Beschränkung des Schnittpunkts durch einen Hüllwürfel

Prinzipiell muss die Ermittlung des Schnittpunkts zwischen der Dreiecksfläche und allen Strahlen (Stixeln) durchgeführt werden. Wegen des hohen Diskretisierungsgrades sind zahlreiche solcher Strahlen im Voxelmodell vorhanden. Um die Berechnung des Schnittpunkts zu beschleunigen, wird die Anzahl der sich an der Berechnung des Schnittpunkts beteiligten Strahlen reduziert. Dazu wird ein Hüllwürfel des Dreieckes eingesetzt. Die sich außerhalb des Würfels befindenden Strahlen können von der Ermittlung des Schnittpunkts ausgeschlossen werden (Bild 4.18).

4.3.2.3 Schnittpunktermittlung

Durch drei Punkte P_1, P_2, P_3 wird eine Ebene im 3D-Raum definiert. Diese Ebene kann mathematisch durch die Punkt-Normalform

$$\vec{n} \bullet \vec{P} + d = 0 \quad (4-26)$$

beschrieben werden, wobei

$$\vec{n} = (P_2 - P_1) \times (P_3 - P_1) \quad (4-27)$$

und

$$d = -\vec{n} \bullet \vec{P}_1 \quad \text{ist.} \quad (4-28)$$

Eine Linie im 3D-Raum lässt sich durch folgenden Zusammenhang in der Punktrichtungsform beschreiben, wobei Q_0 den Anfangspunkt und \vec{w} die Richtung der Linie darstellen.

$$Q_{(\lambda)} = Q_0 + \lambda \vec{w} \quad (4-29)$$

Für Stixel wird \vec{w} jeweils nach Richtung als \vec{i} oder \vec{j} oder \vec{k} definiert.

Wird die Formel (4-29) in die Formel (4-26) eingesetzt, ergibt sich die folgende Beziehung:

$$\vec{n} \bullet Q_0 + \lambda(\vec{n} \bullet \vec{w}) = d \quad (4-30)$$

Weiter erhält man:

$$\lambda = \frac{d - \vec{n} \bullet Q_0}{\vec{n} \bullet \vec{w}} \quad (4-31)$$

Wird die Formel (4-31) in die Formel (4-29) eingesetzt, ergibt sich der Schnittpunkt $Q_s(\lambda)$ zwischen Stixel und Ebene:

$$Q_s = Q_0 + \frac{d - \vec{n} \bullet Q_0}{\vec{n} \bullet \vec{w}} \vec{w} \quad (4-32)$$

4.3.2.4 Ermittlung der Existenz eines Schnittpunktes

Der im Abschnitt 4.3.2.3 berechnete Schnittpunkt zwischen der Dreiecksfläche und dem Strahl befindet sich innerhalb des das Dreieck umhüllenden Würfels. Ob sich der gefundene Schnittpunkt auch wirklich innerhalb des Dreieckes befindet, muss durch eine Kontrolle ermittelt werden. Dazu werden Dreieckskoordinaten eingeführt.

Ein Dreieck mit den drei Eckpunkten P_1, P_2, P_3 und der Fläche A (Bild 4.19) wird durch Angabe der Koordinaten x_i, y_i, z_i ($i = 1, 2, 3$) eindeutig im globalen Koordinatensystem erfasst. Als natürliche

Koordinaten werden Dreieckskoordinaten verwendet. Wird ein beliebiger Punkt P des Dreiecks geradlinig mit den Eckpunkten verbunden, so erscheinen die drei Teilflächen A_1 , A_2 , und A_3 . Die Eckpunkte des Dreieckes sind wie folgt festgelegt:

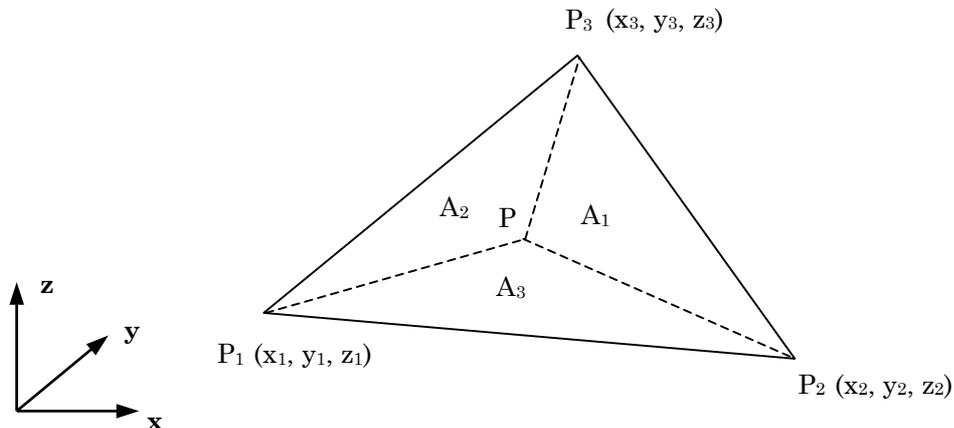


Bild 4.19: Dreieckskoordinatensystem

$$P_1 = \begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix}, \quad P_2 = \begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix}, \quad P_3 = \begin{Bmatrix} x_3 \\ y_3 \\ z_3 \end{Bmatrix} \quad (4-33)$$

Ausgehend von

$$A_1 + A_2 + A_3 = A, \quad (4-34)$$

werden die drei Quotienten

$$L_1 = \frac{A_1}{A}, \quad L_2 = \frac{A_2}{A}, \quad L_3 = \frac{A_3}{A} \quad (4-35)$$

gebildet. Diese werden als die drei Dreieckskoordinaten des Punktes P bezeichnet. Dabei gilt:

$$L_1 + L_2 + L_3 = 1 \quad (4-36)$$

Bild 4.20 zeigt ein Dreieck und einen sich auf dieser Dreiecksfläche befindenden Schnittpunkt P. Zur Beurteilung des Enthaltenseins dieses Schnittpunktes innerhalb des Dreieckes werden folgende drei Schritte durchgeführt:

- Berechnung der Dreiecksflächen: $A(P_1, P_2, P_3)$, $A_1(P, P_2, P_3)$, $A_2(P, P_1, P_3)$, $A_3(P, P_1, P_2)$,

- Berechnung der Dreieckskoordinaten: $L_1 = \frac{A_1}{A}$, $L_2 = \frac{A_2}{A}$, $L_3 = \frac{A_3}{A}$ und
- Beurteilung der Existenz dieses Schnittpunktes: Ist die Bedingung $0.0 \leq L_1, L_2, L_3 \leq 1.0$ nicht erfüllt, so liegt der Schnittpunkt nicht innerhalb der Dreiecksfläche und muss somit nicht weiter betrachtet werden.

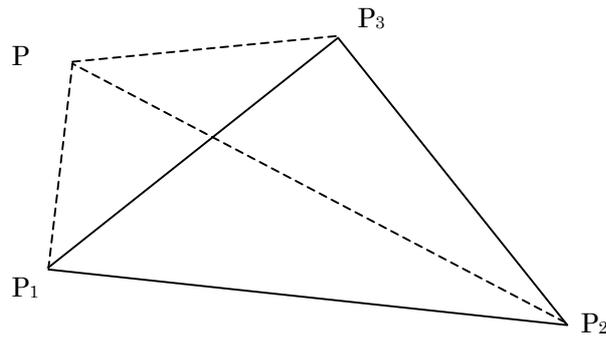


Bild 4.20: Ermittlung der Existenz eines Schnittpunktes

4.4 Gestaltung des voxelbasierten Zerspanungsvolumens

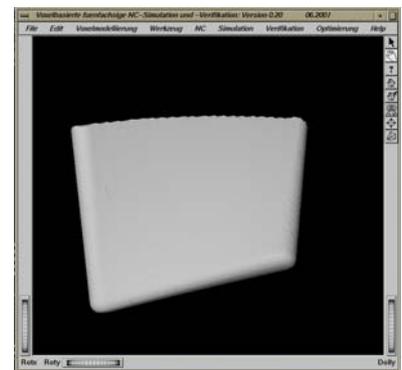
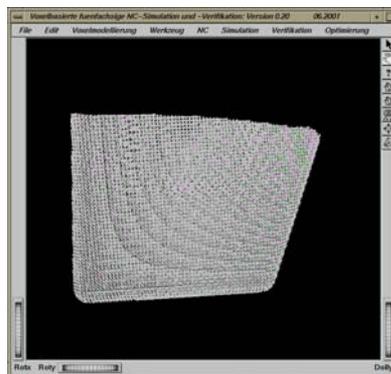
Zur voxelbasierten NC-Simulation wird die NC-Bearbeitung durch Boolesche Subtraktion zwischen voxelbasiertem Materialmodell und Zerspanungsvolumen realisiert. Nach der Diskretisierung der Grundkörper und deren Bewegungsvolumen in Voxelmodelle lässt sich das Zerspanungsvolumen durch die Kombination der diversen Voxelmodelle gestalten. Bild 4.21 stellt die unterschiedlichen voxelbasierten Zerspanungsvolumen dar.

Werkzeugtyp

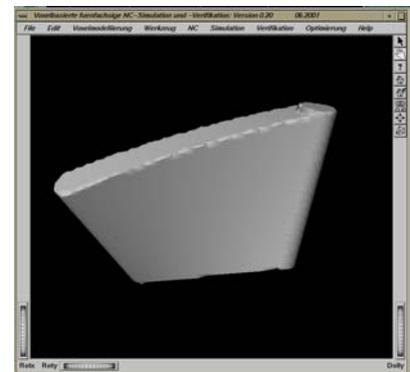
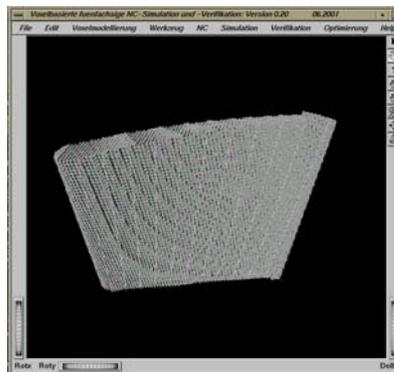
Voxelmodell

Oberflächenmodell

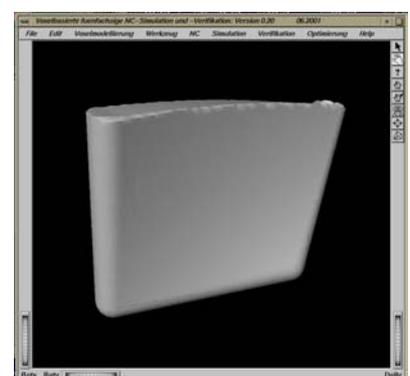
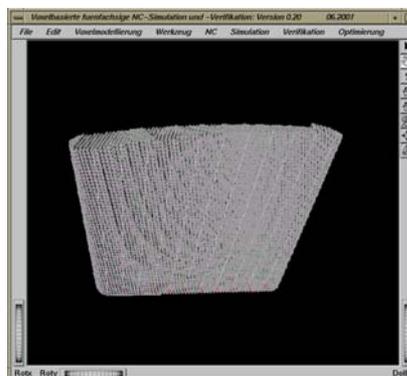
Kugelfräser



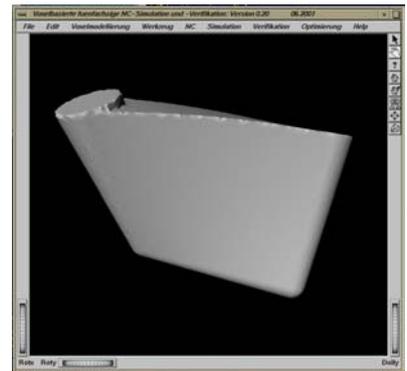
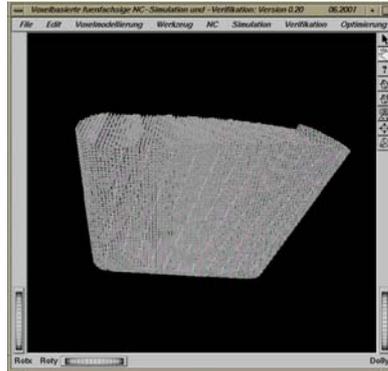
Zylinderfräser



Schaftfräser
mit Eckenradius
(Torusfräser)



Kegeliger Gesenkfräser



Fassfräser

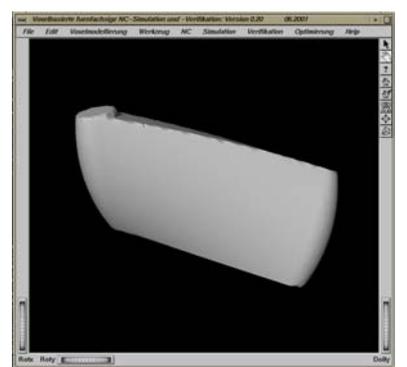
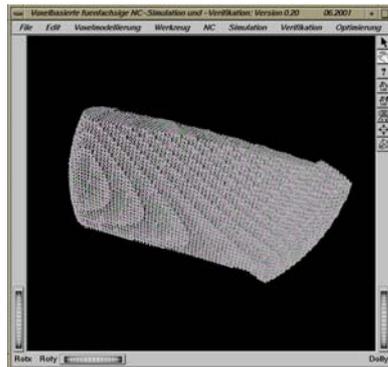


Bild 4.21: Beispiele: Zerspanungsvolumen der diversen Werkzeuge

5 Aktualisierung des simulierten Werkstücks

5.1 Allgemeiner Ablauf der Werkstückaktualisierung

Für eine realistische NC-Simulation hat die Aktualisierung der Werkstückgeometrie eine zentrale Bedeutung. In diesem Kapitel wird zuerst der allgemeine Ablauf der Werkstückaktualisierung gezeigt. Danach werden die verschiedenen Methoden zur Optimierung des Aktualisierungsprozesses beschrieben.

Entsprechend der definierten Fertigungsaufgabe werden Werkzeugmaschinen mit Weg- und Schaltanweisungen angesteuert [91, 92]. Die NC-Steuerinformationen werden mittels NC-Programme digitalisiert. Die NC-Programme werden üblicherweise vom Postprozessor, welcher die Weg- und Schaltinformationen an die spezielle Maschinensteuerung anpasst, an die NC-Steuerung der Werkzeugmaschine übergeben. Danach erfolgt die Interpretation der NC-Programme durch die Steuerung der Werkzeugmaschine. Dabei werden die Weginformationen des NC-Programms in analoge Bewegungen der Maschinenachsen umgewandelt [93, 94, 95].

```
●●●●●●●●
GOTO/-44.6290,17.5055,59.3474,-0.0237347,0.9976376,0.0644654
GOTO/-44.5315,17.5084,59.1954,-0.0267291,0.9972590,0.0689930
GOTO/-44.4340,17.5120,59.0434,-0.0297093,0.9968532,0.0734924
GOTO/-44.3364,17.5162,58.8914,-0.0326751,0.9964206,0.0779639
GOTO/-44.2389,17.5211,58.7394,-0.0356267,0.9959617,0.0824076
GOTO/-44.1414,17.5266,58.5874,-0.0385638,0.9954770,0.0868235
GOTO/-44.0439,17.5327,58.4354,-0.0414866,0.9949670,0.0912119
GOTO/-43.9464,17.5395,58.2835,-0.0443949,0.9944320,0.0955729
GOTO/-43.8490,17.5469,58.1316,-0.0472888,0.9938724,0.0999066
GOTO/-43.7515,17.5550,57.9797,-0.0501682,0.9932889,0.1042132
GOTO/-43.6541,17.5637,57.8278,-0.0530331,0.9926816,0.1084928
GOTO/-43.5566,17.5730,57.6760,-0.0558835,0.9920511,0.1127456
GOTO/-43.4592,17.5829,57.5242,-0.0587194,0.9913978,0.1169718
GOTO/-43.3618,17.5935,57.3724,-0.0615407,0.9907221,0.1211715
●●●●●●●●
```

Bild 5.1: Beispiel: Cutter Location Data (CLDATA)

Analog zur realen Fertigungsaufgabe erfolgt bei der Simulation zuerst die Anpassung der NC-Programme an die spezielle Maschinensteuerung. Anschließend werden sie durch die Steuerung der NC-Maschine interpretiert. Hierzu bedarf es der Kenntnis über die Maschinensteuerung und das Format des NC-Programms. In der vorliegenden Arbeit werden die steuerungsneutralen NC-Anweisungen, die beispielsweise im CLDATA-Format (Bild 5.1) vorliegen [88], direkt in das NC-

Simulationssystem eingesetzt. Bei dieser direkten Kopplung der CLDATA-Anweisungen mit dem NC-Simulationssystem entfällt die Anpassung der NC-Programme an die entsprechende spezielle Werkzeugmaschine und die Interpretation der durch die Anpassung umgesetzten DIN 66025-Daten [96]. Diese Anpassung und die Interpretation der NC-Programme bilden keinen Gegenstand der vorliegenden Arbeit. Sie werden in der weiteren Forschung (VP: Virtuelle Produktion sowie VME: Virtual Manufacturing Environment) [97, 98] behandelt. Bild 5.1 zeigt beispielsweise die in der vorliegenden Arbeit angewendeten CLDATA.

Die Vorbereitung für die voxelbasierte NC-Simulation erfolgt durch das VCM-System (Virtual Clay Modelling). Ausgehend von der vom Konstrukteur vorgegebenen Simulationstoleranz und den Radiusinformationen wie dem Radius eines Kugelfräasers und dem Abrundungsradius eines Torusfräasers wird der Diskretisierungsgrad für ein Voxelmodell mittels der im Kapitel 3 beschriebenen Methoden berechnet. Anschließend wird ein Voxelmodell (Tonmodell: hier ein Würfel) nach den Außenmaßen des zu bearbeitenden Werkstücks im VCM-System definiert. Durch Bearbeitungen wie Materialabtragung, -auftragung und -ausgleich dieses Voxelmodells im VCM-System nach den Außenmaßen des zu bearbeitenden Werkstücks wird das rechnerinterne Materialmodell für die voxelbasierte NC-Simulation hergestellt (siehe Abschnitt 3.5) und dem NC-Simulationssystem übergeben.

Im Bild 5.2 wird der allgemeine Ablauf einer Werkstückaktualisierung in der voxelbasierten fünfachsigen NC-Simulation in Form eines Ablaufdiagramms dargestellt. Damit werden die Hauptaktivitäten in der Werkstückaktualisierung aufgezeigt. Die Systemsteuerung sowie der ganze Simulationsablauf werden in den folgenden Kapiteln ausführlich behandelt.

Nach dem Start wird die Initialisierung des Simulationssystems durchgeführt (Schritt 1). In diesem Schritt wird das im VCM-System generierte Materialmodell und die zu simulierenden NC-Programme (CLDATA) eingegeben, ebenso die Informationen über das eingesetzte Werkzeug. Das Materialmodell und das Werkzeug können graphisch auf dem Bildschirm dargestellt und visuell geprüft werden.

Nach der Initialisierung des Systems wird die dynamische voxelbasierte NC-Simulation ausgeführt. Danach erfolgen die Eingabe des ersten CLDATA-Satzes und die Darstellung des Werkzeuges, entsprechend der im ersten CLDATA-Satz repräsentierten Werkzeugposition und Achsenrichtung, auf dem Bildschirm (Schritt 2). Dabei wird es dem Anwender ermöglicht, das gezeichnete Materialmodell mit dem Zustand des Werkzeugs am Anfang der Bearbeitung zu vergleichen. Üblicherweise soll das Werkzeug zu diesem Zeitpunkt das Materialmodell nicht berühren. Nach diesem Schritt wird

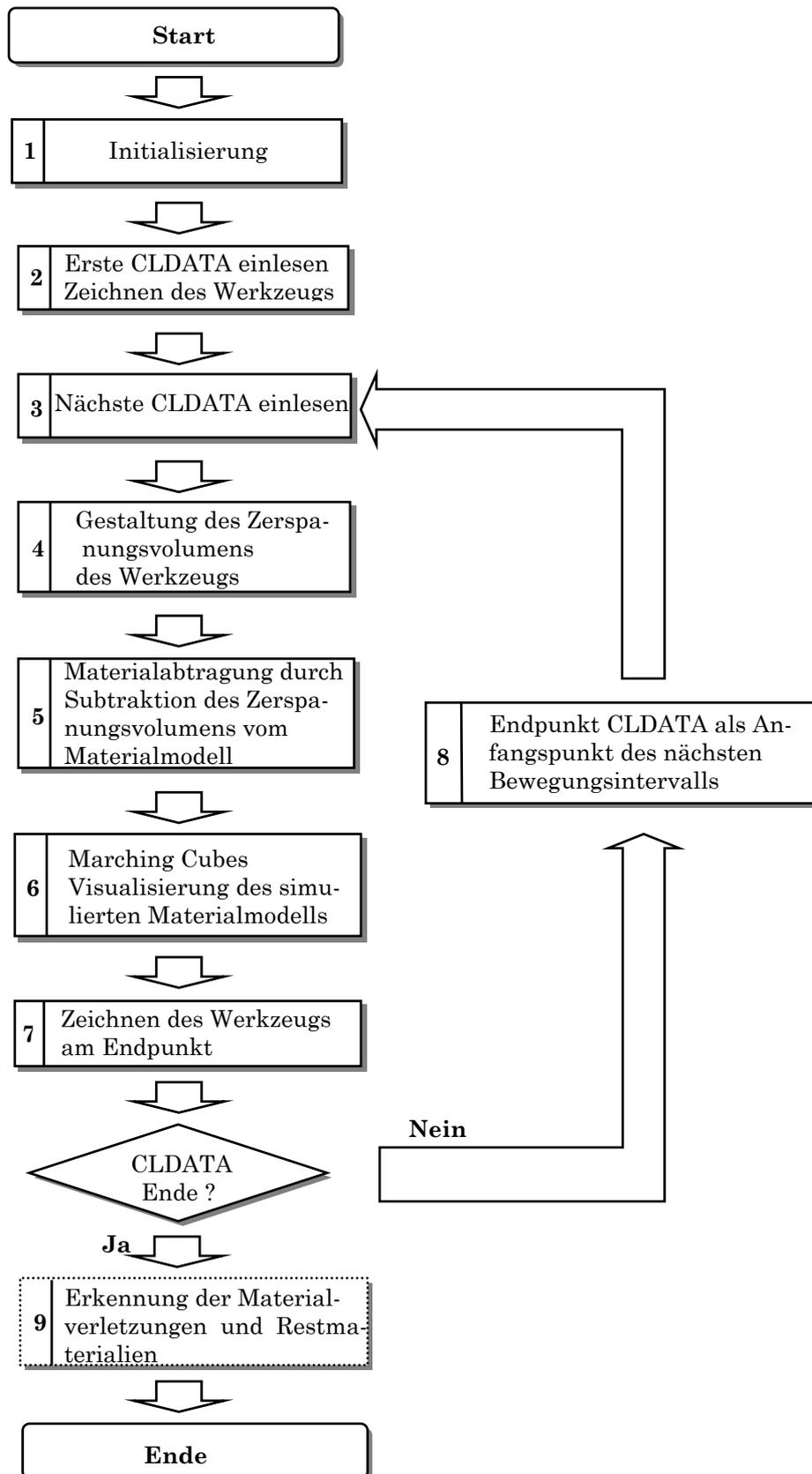


Bild 5.2: Allgemeiner Ablauf in der Werkstückaktualisierung

der nächste CLDATA-Satz eingegeben (Schritt 3).

Im Schritt 4 erfolgt die Ermittlung des Zerspanungsvolumens des Werkzeugs zwischen dem Bewegungsintervall. Nach der Swept-Envelope Methode werden zuerst die Oberflächen des Zerspanungsvolumens mit Dreiecksnetzen annähernd dargestellt. Danach wird das durch die Oberflächen repräsentierte Zerspanungsvolumen ins vom Stixelmodell repräsentierte Voxelmodell umgewandelt. Die Größe des einzelnen Voxels des voxelbasierten Zerspanungsvolumens ist identisch mit der des voxelbasierten Materialmodells. Die Dimension des voxelbasierten Zerspanungsvolumens wird jedoch durch die Funktion Offset durch einen kleineren Raum des voxelbasierten Materialmodells begrenzt. Dies ist für die weitere segmentierte Materialabtragung und Visualisierung hilfreich.

Die Materialabtragung (Schritt 5) erfolgt durch die Subtraktion des voxelbasierten Zerspanungsvolumens des Werkzeugs von dem Materialmodell. Dabei wird die Boolesche Operation impliziert durchgeführt. Um diesen Vorgang beschleunigen zu können, wird die Subtraktion lokal eingegrenzt auf den tatsächlichen Bereich der notwendigen Modelländerung.

Im nächsten Schritt erfolgt die Visualisierung des simulierten Werkstücks. Dabei wird zuerst die Oberfläche des simulierten Werkstücks mittels Marching-Cubes-Verfahrens extrahiert. Dadurch wird eine geschlossene Dreiecksstruktur der Oberfläche des simulierten Werkstücks generiert. Die Darstellung dieser Dreiecksstruktur auf dem Bildschirm erfolgt mittels der Graphikbibliothek Open Inventor, welche auf die GL-Graphikbibliothek aufsetzt und in hohem Maße auf die Graphik-Hardware zugeschnitten ist.

Nach der Visualisierung des simulierten Werkstücks erfolgt die graphische Darstellung des Werkzeugs an der Endposition des Bewegungsintervalls (Schritt 7). Durch die Darstellung des Werkzeugs nach jedem NC-Programmschritt und die Abbildung des simulierten Werkstücks wird dem Anwender der Eindruck einer dynamischen NC-Bearbeitung vermittelt. Enden die NC-Programme (CLDATA) nach Bearbeitung des NC-Programmschritts noch nicht, wird die Endposition als Anfangsposition des nächsten NC-Programmschritts gesetzt (Schritt 8). Danach wird der NC-Bearbeitungsschritt wiederholt.

Am Ende der NC-Simulation erfolgt die Erkennung der Materialverletzungen sowie die Ermittlung der Restmaterialien. Die Methoden dazu werden im folgenden Kapitel detailliert beschrieben.

In diesem Unterkapitel wird der Ablauf der Werkstückaktualisierung in der voxelbasierten fünfschichtigen NC-Simulation dargestellt. In der vorliegenden Arbeit werden geeignete Maßnahmen zur Be-

schleunigung des Aktualisierungsprozesses des simulierten Werkstücks getroffen. Diese Maßnahmen werden in den folgenden Unterkapiteln dargestellt.

5.2 Lokalisierung

Für ein hochaufgelöstes Voxelmmodell ist mit der Abfrage auf Material im gesamten Voxelraum eine enorme Zahl von Voxeln verbunden. Zur Verminderung des Zugriffs auf Voxel werden die Booleschen Operationen zur Materialabtragung und die graphische Darstellung des Voxelmmodells lokal auf den tatsächlichen Bereich der notwendigen Modelländerung begrenzt. Für die Materialabtragung des Zerspanungsvolumens von dem Materialmodell werden die Operationen in zwei Schritte aufgeteilt. Zum ersten ist der Zugriff auf die Materialinformationen in den beiden Voxelräumen und zum zweiten die Subtraktion zwischen beiden Voxelmodellen von Bedeutung. Deshalb stehen in der vorliegenden Arbeit das voxelbasierte Materialmodell und das Zerspanungsvolumen zur Lokalisierung des Zugriffs auf die Materialinformationen und der Materialabtragung im Vordergrund.

Zur Gestaltung des Zerspanungsvolumens in der voxelbasierten fünfachsigen NC-Simulation werden zwei Schritte durchgeführt. Der Erste ist die Gestaltung des dreiecksnetzbasieren Zerspanungsvolumens, der Zweite die Umwandlung dieses Zerspanungsvolumens ins stixelbasierte Voxelvolumen. Diese Umwandlung wird nur begrenzt im Voxelraum durchgeführt (Bild 5.3).

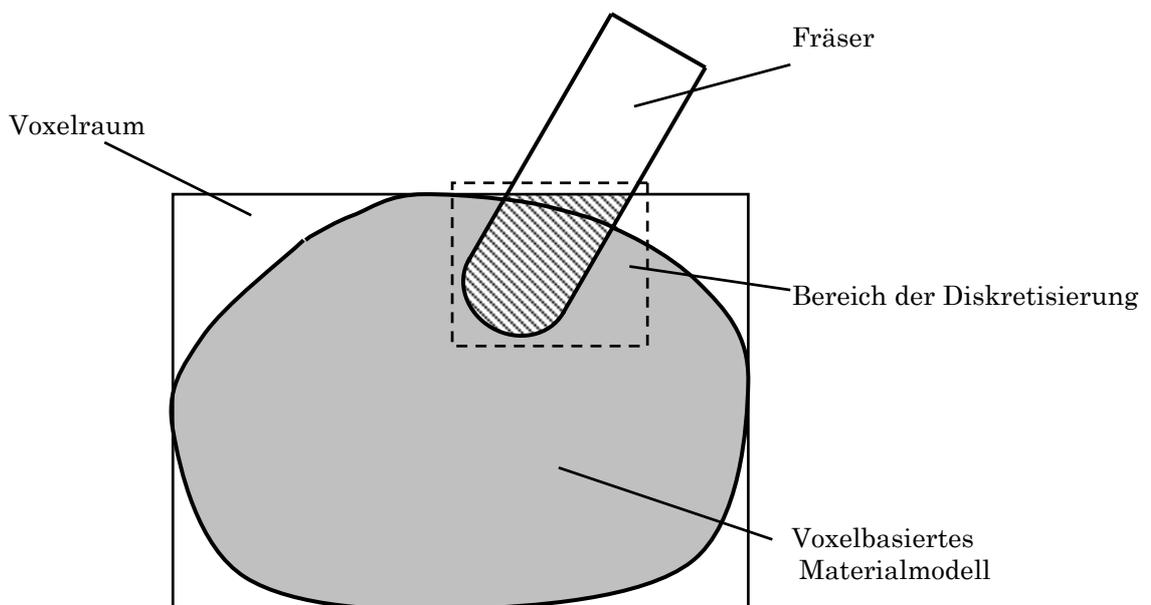


Bild 5.3: Begrenzung der Diskretisierung des Zerspanungsvolumens

Unter einem Voxelraum ist in der vorliegenden Arbeit ein minimaler hexaedrischer 3D-Raum zu verstehen, welcher das voxelbasierte Materialmodell umhüllt. Wie aus Bild 5.3 entnommen werden kann, wird lediglich der im Voxelraum befindliche Teil des Zerspanungsvolumens ins Voxelmodell diskret umgewandelt.

Unter bestimmten Bedingungen wird die Gestaltung des Zerspanungsvolumens des Werkzeugs nicht für jeden Bearbeitungsschritt durchgeführt. Ein Gradienten- und Positionsvergleich der Werkzeuglagen aufeinander folgender Bearbeitungsschritte ermöglicht eine Zusammenfassung dieser Bearbeitungsschritte unter Beachtung ausgezeichneter Randbedingungen. Die Randbedingungen sind für das dreiachsige Fräsen durch kolineare Positionen der „cutter locations“ bestimmt. Für das fünfachsiges Fräsen muss zusätzlich die Orientierung der Längsachse des Werkzeuges mit einbezogen werden. Nach der Zusammenfassung der aufeinander folgenden Bearbeitungsschritte erfolgt einmalig die Gestaltung des dreiecksnetzbasierten Zerspanungsvolumens und dessen Diskretisierung.

Da sich die Verknüpfung sehr kleiner oder sehr großer Voxelvolumen mit dem Werkstückvolumen nachteilig auf die Performanz der Simulation auswirkt, muss die mit dem Gestaltungsbereich verknüpfte Anzahl der Zellen des Zerspanungsvolumens gesteuert werden. Dazu wird ein kleinerer, temporärer Zwischenpuffer eingesetzt. Dieser fasst ein Teil der aufeinander folgenden Bearbeitungsoperationen zusammen. Bei Überschreitung der Grenzen dieses Buffers erfolgt eine Aktualisierung des Werkstückmodells.

Mit wachsender Voxelspeichergröße ist auch ein Divide-and-Conquer-Verfahren für das Materialmodell einzusetzen. Dabei wird eine Unterteilung des untersuchten Darstellungsvolumens angestrebt. Dadurch wird die Anzahl der erforderlichen Diskretisierungen weitestgehend auf die tatsächliche Überdeckung zwischen Materialmodell und Zerspanungsvolumen begrenzt.

5.3 Variable Visualisierungen des simulierten Werkstücks

5.3.1 Ausnutzung der Stixelrepräsentationsform bei der Oberflächenextraktion

Ein wesentlicher Schritt der dynamischen NC-Simulation ist die Darstellung veränderlicher Zustände. Da sich das simulierte Werkstück und die Position des Werkzeuges während der Bearbeitung rasch ändern, ist für eine dynamische NC-Simulation eine ständige Aktualisierung des simulierten

Werkstücks und des bewegten Werkzeugs erforderlich [99]. Bei der voxelbasierten NC-Simulation erfolgt die Visualisierung in folgenden zwei Schritten:

- Oberflächenextraktion des simulierten voxelbasierten Werkstücks mittels Marching-Cubes-Verfahren,
- Aufzeichnung der extrahierten Polygonstruktur seitens der Systemfunktionen der Graphikstation.

Während der Aufzeichnungsvorgang nur über den Umfang der Polygonmenge beeinflusst und anwendungsseitig nicht gesteuert wird, kann die Oberflächenextraktion des simulierten Werkstücks selektiv erfolgen und ist daher der Ansatzpunkt zur Reduktion des Darstellungsaufwandes.

Da die Voxeldarstellung durch die Stixel-Repräsentation in einer Art Lauflängen-Kodierung vorliegt, lässt sich in zumindest einer Laufrichtung der Effekt des Überspringens leerer Bereiche realisieren. Die Anzahl der Materialüberprüfungen wird dadurch minimiert.

Anhand der Lauflängen-Kodierung kann man bereits beim ersten Zugriff erkennen, ob sich in der aktuell untersuchten Laufrichtung überhaupt ein Materialwechsel befindet. Ist dies nicht der Fall, können zwei der acht zur Triangulation benötigten Vertices für diese Zeile konstant als „außerhalb“ des Objektes gehalten werden. Gilt dies für alle vier Stixel, die zur Ermittlung der Triangulations-Vertices nötig sind, kann diese Zeile übersprungen werden. Daher müssen in leeren Segmenten nicht mehr n^3 Voxelpositionen untersucht werden, sondern nur n^2 .

Befinden sich in einer Zeile nur wenige Materialwechsel, kann ebenfalls eine erhebliche Reduzierung der Untersuchungsschritte erreicht werden. Statt sich wie beim Zellmodell mit einer konstanten Schrittweite durch den Datensatz zu bewegen, werden die ersten Materialwechsel-Position aus den vier benötigten Stixeln direkt ermittelt. Danach lässt sich die Materialüberprüfung wie beim Zellmodell durchführen.

5.3.2 Segmentierte Visualisierung

Da die Generierung der gesamten Oberfläche des simulierten Werkstücks nach jedem Bearbeitungsschritt viel zu viel Zeit in Anspruch nimmt, ist der Voxelraum in viele kleine Segmente unterteilt (Bild 5.4). Jede am Werkstückmaterialmodell vorgenommene Materialabtragung liefert deshalb die Eckpunkte des umschließenden Würfels an den Marching-Cubes, der diese Koordinaten dann auf die festen, internen Segmentgrenzen umrechnet. Der Marching-Cubes ist dadurch in der Lage, die Segmente unabhängig voneinander zu berechnen. Solange die Schrittweite in allen Segmenten konstant

ist, ist gewährleistet, dass ein geschlossenes Modell entsteht. Dies begründet sich daraus, dass die Schnittpunkte des Objektes mit den Stixeln zwischen zwei benachbarten Zellen an der gemeinsamen Fläche gleich sind. Somit existiert ein kontinuierlicher Übergang zwischen den Segmenten. Die Ausführung der Oberflächenextraktion des simulierten Werkstücks kann selektiv in den entsprechenden Segmenten erfolgen.

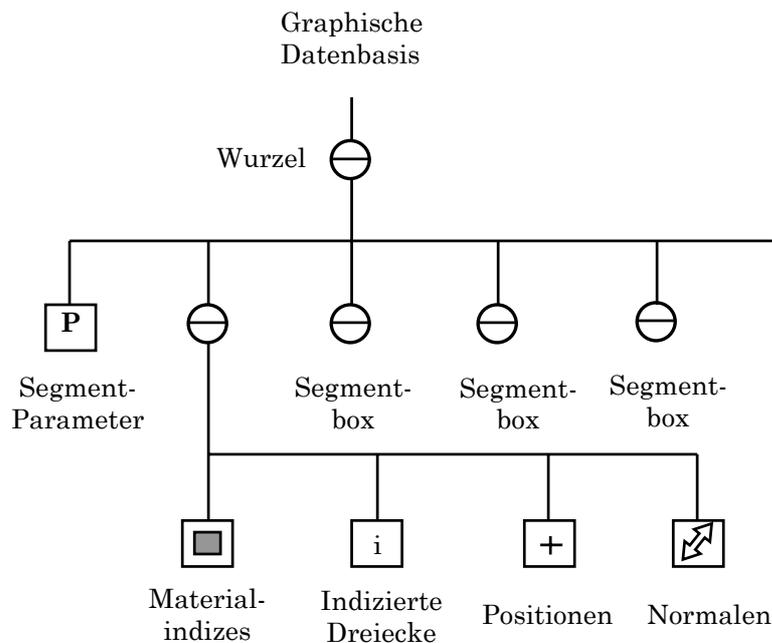


Bild 5.4: Segmentorientierte graphische Datenbasis [24]

5.3.3 Darstellungsverfahren durch variable Auflösung

Im allgemeinen lässt sich das Materialmodell für die NC-Bearbeitung in einfacher Form wie Würfel oder Zylinder gestalten. Während der bearbeitete Bereich des Materialmodells mit der NC-Bearbeitung immer komplizierter wird, bleibt der übrige Teil des Materialmodells unverändert. Für eine dynamische NC-Simulation ist eine stetige Aktualisierung des simulierten Werkstücks erforderlich. Eine Verringerung des Aufwandes bei der Visualisierung des simulierten Werkstücks kann durch eine selektive Auswertung der Segmente des Voxelmodells erreicht werden.

Obwohl das rechnerinterne voxelbasierte Werkstückmodell stets mit dem gleichen Diskretisierungsgrad repräsentiert und aktualisiert wird, ist eine graphische Darstellung mit unterschiedlichen Auflösungen möglich. Das adaptive Marching-Cubes-Verfahren erlaubt die gute Objekt-Approximation des Marching-Cubes-Verfahrens bei gleichzeitiger erheblicher Reduktion der Anzahl der Oberflächen-Dreiecke [100, 101]. Die Reduzierung der Dreiecke wird dadurch erreicht, dass der

unbearbeitete Bereich des Werkstückmodells mit einem Datensatz geringerer Auflösung extrahiert wird.

Der Umstand, dass der Volumen-Datensatz nicht mehr in Segmente homogener Größe unterteilt wird, führt zu dem Problem, dass Löcher in der Dreiecksstruktur entstehen können. Es bilden sich Löcher an Stellen, an denen Segmente unterschiedlicher Größe aneinander grenzen.

Um nun die Löcher zwischen den verschiedenen Segmenten schließen zu können, müsste eine globale Datenstruktur aufgebaut werden. Dies wäre jedoch nur mit einem erheblichen Aufwand an Datenstruktur und Rechenzeit zu realisieren. Weiterhin würde die ursprünglich vorhandene Parallelisierbarkeit verloren gehen, da alle Segmente auf nur eine Datenstruktur zugreifen müssten.

Es bietet sich an, dass der Anwender auswählen kann, ob und wann diese Methode eingesetzt wird.

5.3.4 Gesteuerte Visualisierung

Eine unmittelbare Verminderung des Aufwandes der voxelbasierten NC-Simulation kann durch eine steuerbare Visualisierung erfolgen. Die NC-Simulationsprozesse werden bei jedem Bearbeitungsschritt geometrisch vollständig ausgeführt. Das simulierte Werkstück ist jedoch nicht notwendigerweise nach jedem Bearbeitungsschritt graphisch darzustellen. Wichtig dabei ist, dass NC-Programmfehler durch die NC-Simulation erkannt werden, und Materialverletzungsbereiche im simulierten Werkstück aufgezeichnet werden. Durch kontrollierte Darstellung und Ausblenden kann eine Verbesserung des Laufzeitverhaltens erreicht werden.

Der Anwender kann die Schrittweite für Visualisierungen entweder über eine Systemfunktion definieren oder durch Scrollen des NC-Programms mit Hilfe einer Bildlaufleiste, in Anlehnung an gängige Fensterlayouts im Bereich der Window-Technik (Windows, X-Windows [102, 103], Motif), die Visualisierungsposition auswählen. Eine Visualisierung erfolgt nur dann, wenn ein NC-Programmfehler sowie eine Materialverletzung vorliegen oder eine definierte Visualisierungsposition (NC-Zeile) erreicht wird. Hierbei findet keine kontinuierliche graphische Ergebnisdarstellung statt. Der NC-Simulationsprozess wird damit drastisch beschleunigt.

Während der NC-Simulation kann lediglich das simulierte Werkstück graphisch durch Voxeln aufgezeichnet werden. Dabei entfällt die zeitaufwendige Oberflächenextraktion des voxelbasierten Werkstücks und deren Aufzeichnung. Die direkte Visualisierung des voxelbasierten Werkstücks erfolgt mittels der Graphikbibliothek „Open Inventor“, die auf die GL-Graphikbibliothek aufsetzt und in hohem Maße auf die Graphik-Hardware zugeschnitten ist. Das hat eine höhere Beschleuni-

gung des Simulationsprozesses zur Folge. Jedoch verschlechtert sich dabei die Oberflächenqualität des dargestellten Werkstücks (Bild 5.5). Am Schluss der NC-Simulation kann die Oberfläche des simulierten Werkstücks realitätsnah nachträglich dargestellt werden.

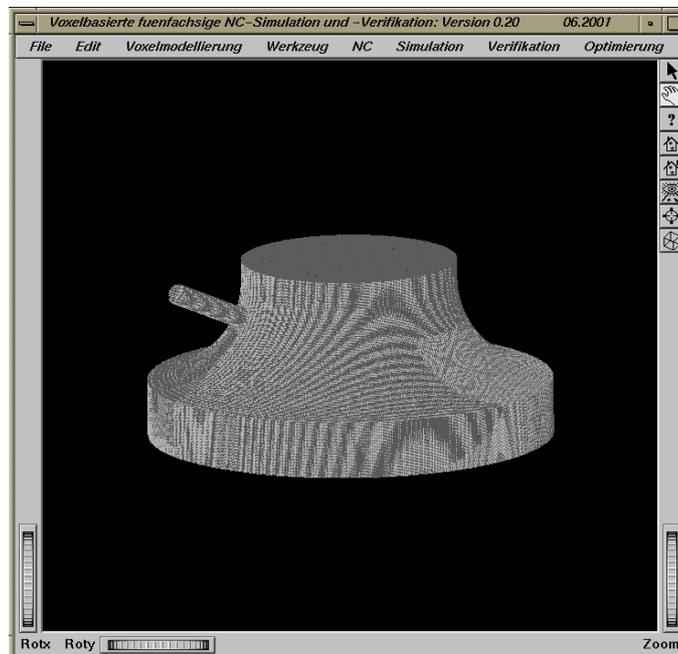


Bild 5.5: Voxelbasierte graphische Darstellung der NC-Bearbeitung

Der Vorgang der Visualisierung des simulierten Werkstücks kann in der vorliegenden Arbeit während der Simulation ausgeblendet werden. Die NC-Simulation erfolgt nur geometrisch durch Aktualisierung des rechnerinternen voxelbasierten Werkstücks. Informationen der NC-Simulation wie NC-Programmfehler und Materialverletzungsbereiche werden zuerst in der Systemstruktur abgelegt. Nach Beendigung der NC-Simulation sowie dem Auftreten eines Fehlers lassen sich das simulierte Werkstück darstellen und die Fehler in den entsprechenden Bereichen im simulierten Werkstück hervorheben.

5.4 Konzept der Parallelisierung

Ein weiterer wichtiger Aspekt zur Beschleunigung der voxelbasierten NC-Simulation ist die Parallelisierung auf Basis eines Parallelrechners. Bei der voxelbasierten NC-Simulation gewährt ein normaler Rechner keine ausreichende Rechnerunterstützung für eine echtzeitnahe Durchführung, da zahlreiche Gestaltungen von Zerspanungsvolumen und Mixoperationen zwischen zwei Voxelmolelen sowie häufige Zugriffe auf den Voxelraum zeitaufwendig sind, und somit der NC-

Simulationsvorgang verzögert wird. Daher muss auf herkömmliche Methoden zurückgegriffen werden, um eine echtzeitnahe NC-Simulation erreichen zu können. Erst durch den Einsatz von Parallelrechnern kann die voxelbasierte NC-Simulation zufriedenstellend ausgeführt werden.

Einfachere Algorithmen, geringe Datenkomplexität und sehr große Datenmengen prädestinieren die voxelbasierten Methoden für eine Parallelisierung. Durch die Komprimierung des Voxelmodells mittels der Stixel-Repräsentation und die dabei erreichte Vermeidung von Datenabhängigkeiten wird die Parallelisierung der Mixoperationen, welche eine Verknüpfung des Zerspanungsvolumens mit dem Materialvolumen verwirklichen, in Abhängigkeit von der zur Verfügung stehenden Hardware ermöglicht.

Bei der Betrachtung des Ablaufs der Simulationsalgorithmen und der Strukturierung der Ablaufschritte des Voxelmodells stellt man fest, dass alle wichtigen Schritte in Bezug auf die eingesetzten geometrischen Verfahren in einfacher Weise parallelisierbar sind (Bild 5.6).

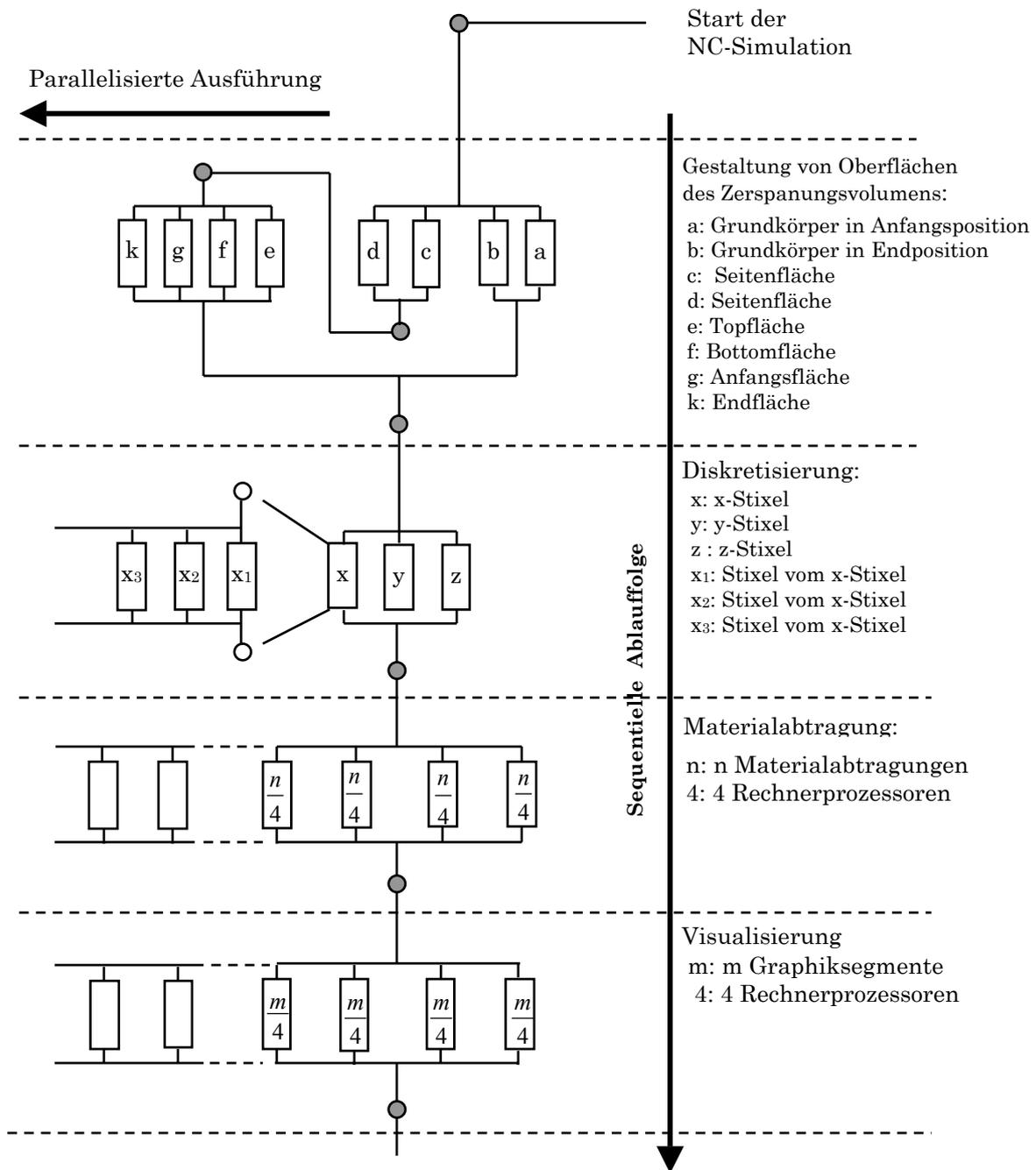


Bild 5.6: Parallelisierte Ablaufschritte der voxelbasierten NC-Simulation
(Erweitertes Bild aus [24])

Eine Parallelisierung des Ablaufs ist in den folgenden Ablaufschritten möglich:

- Berechnung des dreiecksbasierten Bewegungsvolumens: Die Gestaltungen des Grundkörpers des Werkzeugs an der Anfangs- und Endposition des Bewegungsintervalls und auch der zwei Seitenflächen des Bewegungsvolumens werden voneinander unabhängig und parallel durchgeführt. Nach der Ermittlung der zwei Set-kritischen Kurven bei der Gestaltung der zwei Seitenflächen werden die Bedingungen für die weitere Parallelisierung der Modellierung von Top-, Bottom-, Anfangs- und Endflächen des Bewegungsvolumens geschaffen.
- Diskrete Umwandlung des oberflächenbasierten Zerspanungsvolumens in ein voxelbasiertes Zerspanungsvolumen: Die Voxeldarstellung durch Stixel-Repräsentation ist sowohl je Koordinatenrichtung gesondert als auch für eine Koordinatenrichtung weiter unterteilbar. Neben der parallelen Diskretisierung des oberflächenbasierten Zerspanungsvolumens in X-, Y- und Z-Koordinatenrichtungen sind die Potenziale der Parallelisierung der Diskretisierung in jeder Koordinatenrichtung vorhanden.
- Materialabtragung des Zerspanungsvolumens von dem Werkstückmaterialmodell: Durch die Vermeidung von Datenabhängigkeiten des Voxelmodells mit Hilfe der Stixel-Repräsentation sind alle Möglichkeiten für weitgehende Parallelisierung der Materialabtragung gegeben. Konkret bedeutet dies, dass die Materialabtragung intern auf eine Mischoperation zwischen zwei Instanzen des voxelbasierten Volumenmodells abgebildet wird. Diese Mischoperation setzt sich aus den Mischoperationen der einzelnen Stixel der beiden Voxelmodelle zusammen, zwischen denen keine Datenabhängigkeit besteht. Dadurch, dass die Mischoperationen der Stixeln parallel zueinander ausgeführt werden, kann ein hohes Maß an paralleler Ausführung erreicht werden.
- Visualisierung des voxelbasierten simulierten Werkstückmodells: Da die Generierung der gesamten Oberfläche im Werkstückmodell nach jedem Bearbeitungsschritt viel zu viel Zeit in Anspruch nehmen würde, wird der Objektraum in viele kleine Segmente unterteilt. Das Visualisierungsverfahren bietet über die Segmentstruktur die Möglichkeit, Subvolumen gesondert zu aktualisieren, und an die Verfügbarkeit von Prozessoren anzupassen.

Die Parallelisierung der weiteren Schritte folgt aus den Eigenschaften des eingesetzten Repräsentationsschemas. Lediglich der Prozess der Aufzeichnung ist in gängigen Graphiksubsystemen anwendungsseitig nicht zu steuern. Rechnerarchitekturen mit parallelverarbeitendem Graphiksystem bieten bereits jedoch auch für diesen Schritt eine weitgehende Hardwareunterstützung an.

Eine Untersuchung mit einem 4-Prozessor-Rechner und parallelisierenden Übersetzerprogrammen ergab einen zur Anzahl zugeschalteter Prozessoren proportionalen Anstieg der Rechengeschwindigkeit für die Zwischenschritte der Diskretisierung und der Materialabtragung. Dieses lineare Verhalten kann dahingehend erklärt werden, dass mit massivparallelisierenden Rechnerarchitekturen oder Methoden der verteilten Parallelisierung eine enorme Leistungssteigerung der Berechnungen für die echtzeitnahe voxelbasierte NC-Simulation möglich ist.

Eine vollständige Parallelisierung der voxelbasierten NC-Simulation wie die Berechnung dreiecksbasierter Bewegungsvolumen und die Visualisierung des voxelbasierten Werkstückmodells ist für eine zukünftige Arbeit vorgesehen.

6 Methoden zur Erkennung der Materialverletzungen und Ermittlung der verbliebenen Restmaterialien

6.1 Allgemeine Vorgehensweise bei der voxelbasierten Verifikation des NC-Programms

Eine wichtige Funktion des NC-Simulationssystems ist die Erkennung von NC-Bearbeitungsfehlern während der dynamischen Bewegung der Werkzeugmaschinenachsen. Ein fehlerhaftes NC-Programm kann Kollisionen zwischen den an NC-Bearbeitung beteiligten Elementen wie Spannmitteln, Maschinentisch und Werkzeug oder die Materialverletzungen des Werkstücks auslösen. Da die Elemente der Werkzeugmaschine durch die Kombination von geometrischen Grundkörpern wie Würfel, Zylinder, Kugel, Kegel und Torus in kommerziellen CAD-Systemen modelliert werden können [2, 104, 105] und auch während der dynamischen NC-Bearbeitung unverändert bleiben, kann eine Kollisionsprüfung zwischen den Werkzeugmaschinenelementen im CAD-System durch entsprechende Boolesche Operationen durchgeführt werden. Deshalb liegt eine Kollisionsprüfung außerhalb der vorliegenden Arbeit. In dieser Arbeit wurde sich insbesondere auf die Prüfung der Materialverletzungen und Restmaterialien des simulierten Werkstücks konzentriert.

In diesem Kapitel wird zuerst die Vorgehensweise bei der voxelbasierten Verifikation des NC-Programmes dargestellt. Danach werden die Methoden zur Erkennung der Materialverletzungen des simulierten Werkstücks beschrieben. Am Schluss werden die Methoden zur Ermittlung der Restmaterialien vorgestellt.

Um NC-Bearbeitungsfehler erkennen zu können, müssen die simulierten Flächen mit Soll-Flächen verglichen werden. In der vorliegenden Arbeit wird das simulierte Werkstück durch ein Stixelmodell dargestellt. Die Stixelpunkte liegen genau auf den Oberflächen des simulierten Werkstücks. Die Ermittlung der Abweichung zwischen der simulierten Fläche und der Soll-Fläche kann direkt durch die Berechnung der Abstände zwischen jedem Stixelpunkt und der Sollfläche durchgeführt werden. Bild 6.1 zeigt diese Verifikationsmethode.

Für die Berechnung des Abstandes d zwischen dem aktualisierten Stixelpunkt C und der Soll-Fläche sind unterschiedliche Methoden in der Literatur [106, 107, 108] zu finden. Im allgemeinen wird bei diesen Methoden davon ausgegangen, dass der minimale Abstand zwischen einem Punkt \vec{P} im 3D-Raum und einer freigeformten Fläche \vec{S} am Flächenpunkt \vec{Q} , an dem der Vektor $(\vec{P} - \vec{Q})$ senkrecht zur Tangentenebene der Fläche steht, berechnet wird. Die Symbole des Abstandes d werden durch die Richtung der Flächennormale am Flächenpunkt \vec{Q} bestimmt. Zur Berechnung des Abstan-

des d mit diesen Methoden ist jedoch eine nichtlineare Formel zu lösen. Hier ist üblicherweise ein „Newton/Raphson Search Procedure“ erforderlich [7].

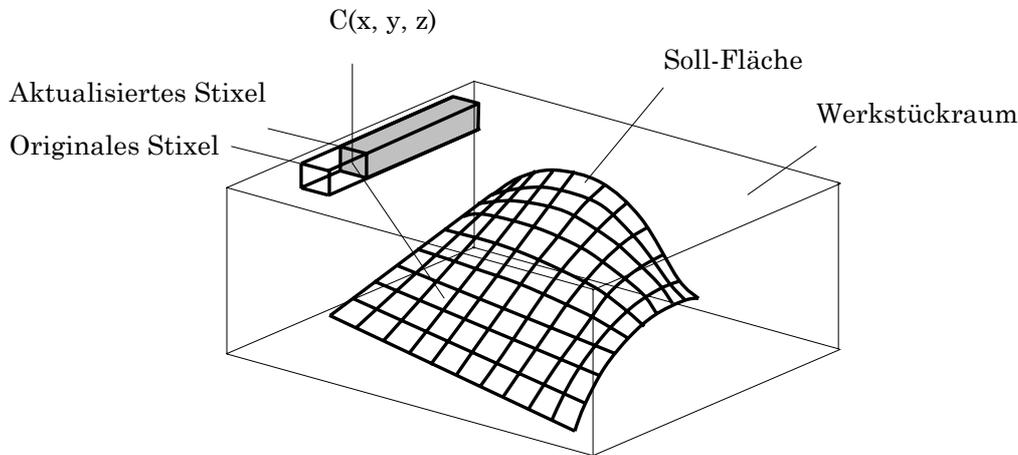


Bild 6.1: Methode zur Verifikation des NC-Programmes [7]

Da die Robustheit und die Effizienz der Newton/Raphson-Methode von der Auswahl des anfänglichen Punkts abhängen [7], wird eine Voxel-Struktur der gesamten Werkstückflächen zur Lieferung der anfänglichen Flächenpunkte für einen gegebenen Stixelpunkt hergestellt. Die Voxel-Struktur kann auch zur Lokalisierung der Flächen des Werkstücks für die Berechnung des Flächenpunkts eingesetzt werden, da nur die Flächen, die dem gegebenen Stixelpunkt am nächsten sind, in der Voxel-Struktur gespeichert werden. Der Flächenpunkt wird in der vorliegenden Arbeit als ein Punkt bezeichnet, der auf einer Fläche des Werkstücks liegt und dem gegebenen Stixelpunkt am nächsten ist. Deshalb wird der minimale Abstand zwischen dem gegebenen Stixelpunkt und diesen Flächen berechnet.

Zur Herstellung der Voxel-Struktur werden zuerst die Soll-Flächen des Werkstücks im 3D-Raum durch ein Stixelmodell unter Berücksichtigung des Flächenfehlers (siehe Abschnitt 3.4) diskretisiert⁴ (Bild 6.2). Die Soll-Flächen des Werkstücks werden durch die von Stixelpunkten gebildeten Dreiecksnetze angenähert. Jedes Dreieck der Dreiecksnetze wird in die Voxel-Struktur [109, 110], die in einen Grenzwürfel des Werkstücks im Objektkoordinatensystem gelegt wird, eingefügt. Die mit Dreiecken durchschnittenen Voxeln werden um den Soll-Flächenindex und den Flächenparametern $[u, v]$ entsprechend des Stixelpunkts erweitert. Da in einem Voxel mehr als ein Stixelpunkt

⁴ Die Berechnungen greifen auf Funktionen des Geometrie Modellierkerns (ACIS) der Firma Spatial Technologies zurück.

erfasst werden kann, wird ein Head-Zeiger einer Punktliste zur Registrierung eingesetzt. Damit kann auf die Stixelpunkte (Flächenpunkte) innerhalb eines gegebenen Voxels zugegriffen werden.

Zur Vervollständigung der Definition der Voxel-Struktur werden weiter zwei zusätzliche Parameter, die Voxelgröße und der Projektionsbereich eingesetzt. Die Voxelgröße beeinflusst direkt die Effizienz der Berechnung des Flächenpunkts, d.h. je größer das Voxel ist, desto mehr anfängliche Flächenpunkte enthält es. Die Genauigkeit der Berechnung des Flächenpunkts wird gewöhnlich nicht von der Voxelgröße beeinflusst. Die minimale Voxelgröße wird durch die Formel (3-11) bestimmt. Je nach der Größe des Werkstücks kann man die tatsächliche Voxelgröße zwischen dem Ein - Fünffachen der minimalen Voxelgröße auswählen.

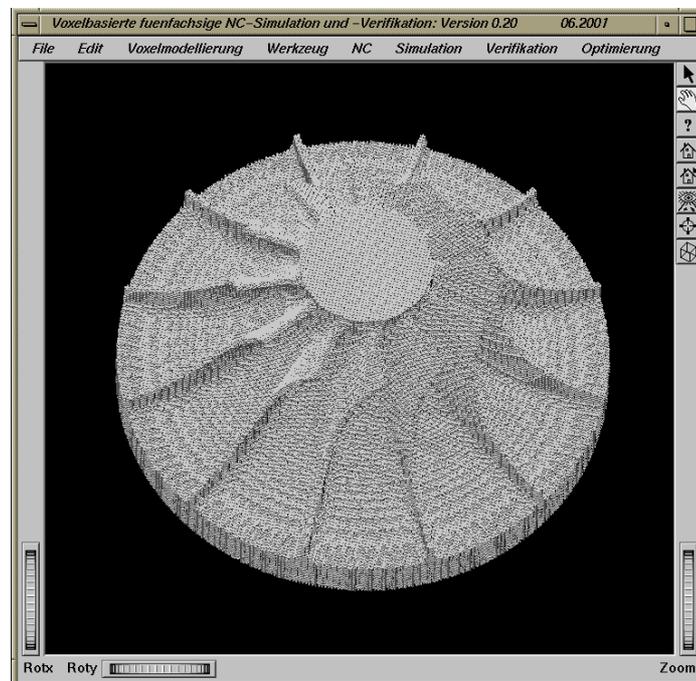


Bild 6.2: Voxelrepräsentiertes Soll-Werkstückmodell

Der Projektionsbereich legt die obere und untere Offset-Grenze vom Dreiecksnetz fest, und somit werden nur die Punkte, die sich innerhalb des Projektionsbereichs befinden, bei der Berechnung des Flächenpunkts betrachtet. Zur Einführung des Projektionsbereichs in die Voxel-Struktur werden die zusätzlichen Voxeln auf beiden Seiten jedes Dreiecks mit entsprechenden Oberflächeninformationen gefüllt, wenn sie innerhalb des Projektionsbereichs liegen (gemessen in Bezug auf die Dreiecksnormale). Zur Bedeckung der konkaven und konvexen Flächenbereiche, die bei der Diskretisierung verloren gehen, muss der Projektionsbereich größer als der Flächenfehler gewählt werden.

Die Voxel-Struktur wird mit einem Voxel-Index (I_x, I_y, I_z) , der durch drei ganze Zahlen repräsentiert wird, kodiert. Die Berechnung des Flächenpunkts erfolgt mit jedem Stixelpunkt in der Punkte-Liste in dem entsprechenden Voxel. Der minimale Abstand unter allen Lösungen wird als NC-Bearbeitungsfehler bezeichnet.

6.2 Erkennung von Materialverletzungen

In dieser Arbeit werden für die NC-Bearbeitung die vom Konstrukteur vorgegebene Toleranzen $[\varepsilon_l, \varepsilon_h]$ verwendet. Diese Toleranzen stellen bei der NC-Programmierung die Untergrenze und Obergrenze dar, welche nicht überschritten bzw. unterschritten werden dürfen. Wenn ein NC-Bearbeitungsfehler die Untergrenze ε_l unterschreitet, tritt eine Materialverletzung auf.

Für die Prüfung der Materialverletzungen im Simulationssystem wurden folgende Betriebsarten realisiert:

- Simulationsbegleitende Prüfung der Materialverletzungen und
- Prüfung der Materialverletzungen im Batch-Betrieb.

Bei der simulationsbegleitenden Prüfung der Materialverletzungen werden die Durchdringungen des Werkzeugs mit den Soll-Flächen nach jedem Bewegungsintervall berechnet. Wegen der Komplexität dieses Vorgangs wird in der vorliegenden Arbeit die im Abschnitt 6.1 vorgestellte Methode eingesetzt. Statt der Berechnung der Durchdringungen des Werkzeugs mit den Soll-Flächen werden die Oberflächen des simulierten Werkstücks mit Soll-Flächen verglichen. Die Prüfung der Materialverletzungen kann auf den jeweils aktualisierten Bereich des Werkstücks begrenzt werden. Die ermittelten Materialverletzungen werden in dem entsprechenden Stixelmodell des simulierten Werkstücks gespeichert. Da ein Stixelmodell in der NC-Simulation mehrmals geschnitten werden kann, werden die Informationen über die ermittelten Materialverletzungen in diesem Stixelmodell nach jedem Schnitt des Stixelmodells aktualisiert.

Die zweite Betriebsart zur Prüfung der Materialverletzungen ist der Batch-Betrieb. Hier findet keine kontinuierliche graphische Ergebnisdarstellung statt. Die Prüfung der Materialverletzungen wird dann durchgeführt, wenn einmal die ausgewählten NC-Zeilen erreicht wurden oder die NC-Simulation komplett abgeschlossen worden ist. Zur Prüfung der Materialverletzungen werden die Abweichungen zwischen den Oberflächen des simulierten Werkstücks und den Soll-Flächen berechnet. Dazu wird auch die im Abschnitt 6.1 vorgestellte Methode verwendet. Die ermittelten Materialverletzungen werden ebenfalls in dem entsprechenden Stixelmodell des simulierten Werkstücks

gespeichert. Die NC-Zeilen, in denen eine Materialverletzung ermittelt wurde, werden in einem Materialverletzungsprotokoll gespeichert. Dieses Protokoll kann nach dem Ablauf des gesamten NC-Programms oder ausgewählter Programmpassagen die Grundlage einer erneuten Simulation bilden, in der die entsprechenden Programmbereiche genauer untersucht werden können.

Dieses Verfahren bietet zwei Vorteile. Zum einen muss der Simulationsprozess nicht fortwährend betrachtet werden. Zum anderen wird ohne kontinuierliche graphische Ausgabe eine Verbesserung des Laufzeitverhaltens erreicht. Dadurch können bei gleicher Berechnungszeit mehr NC-Programme simuliert und verifiziert werden als bei der simulationsbegleitenden graphischen Simulation. Eine Kombination beider Betriebsarten ist möglich.

Vom System erkannte Materialverletzungen können dem Anwender auf unterschiedliche Art und Weise mitgeteilt werden. Das Spektrum der Ergebnisrepräsentation reicht von der Meldung, dass Materialverletzungen erkannt wurden über die Spezifikation der NC-Zeile bis hin zur Berechnung und Darstellung der Materialverletzungsbereiche. In der vorliegenden Arbeit wird die Materialverletzung durch rote Farbe markiert. Die ermittelten Materialverletzungen sind in dem entsprechenden Stixelmodell gespeichert. Bei der Oberflächenextraktion werden diese Informationen weiter von den erzeugten Dreiecken vererbt. Das Dreieck, welches den verletzten Stixelpunkt enthält, wird mit roter Farbe gekennzeichnet. Bild 6.3 zeigt ein Beispiel.

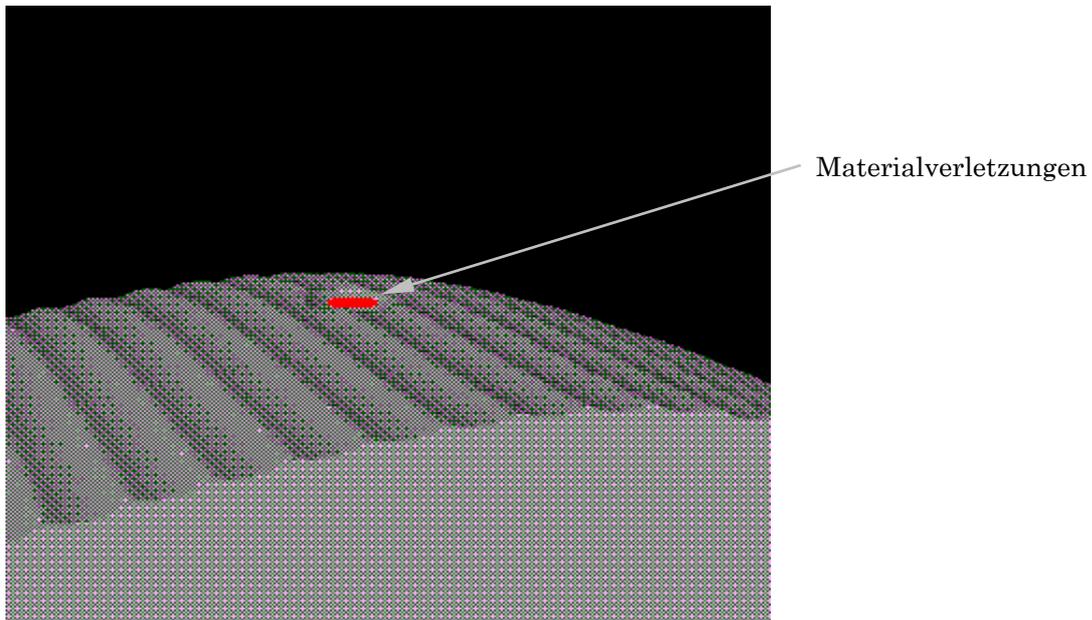


Bild 6.3: Darstellung von Materialverletzungen

6.3 Ermittlung von verbliebenen Restmaterialien

Eine andere wichtige Funktion der NC-Simulation ist die Kontrolle von Restmaterialien, welche nach der NC-Fräsbearbeitung auf Oberflächen des gefrästen Werkstücks bleiben. Die Restmaterialien beeinflussen nicht nur die Oberflächenqualität des Werkstücks, sondern auch die manuelle Nacharbeit, wodurch die verbleibenden Restmaterialien nach der NC-Fräsbearbeitung beseitigt werden. Um die Produktqualität zu verbessern und auch die Nacharbeit möglichst gering zu halten, müssen die Restmaterialien in der NC-Programmierung beachtet werden. Die voxelbasierte NC-Simulation bietet die Möglichkeit, die Restmaterialien auf den Oberflächen des gefrästen Werkstücks zu ermitteln und die entsprechende Informationen zur Optimierung des NC-Programms bereitzustellen.

In dieser Arbeit wird für die NC-Bearbeitung die vom Konstrukteur vorgegebenen Toleranzen $[\varepsilon_l, \varepsilon_h]$ verwendet. Diese Toleranzen stellen für die NC-Programmierung die Untergrenze und Obergrenze dar, welche nicht überschritten werden dürfen. Falls die Restmaterialien die Obergrenze ε_h überschreiten, müssen diese Informationen in der Arbeitsvorbereitung zurückgeführt und dort die geeigneten Maßnahmen getroffen werden.

Die Ermittlung der Restmaterialien wird nach der NC-Simulation durchgeführt. Dabei werden die Oberflächen des simulierten Werkstücks mit den Soll-Flächen verglichen. Dabei wird die im Abschnitt 6.1 vorgestellte Methode verwendet. Die ermittelten Restmaterialinformationen werden ebenfalls in dem entsprechenden Stixelmodell des simulierten Werkstücks gespeichert. Die maximalen Restmaterialien werden in einem Restmaterialprotokoll mit entsprechenden Flächeninformationen hinterlegt. Dieses Protokoll kann nach der Prüfung der Restmaterialien die Grundlage für die Optimierung der NC-Programme bilden, in der die entsprechenden Programmbereiche manipuliert werden können.

In der vorliegenden Arbeit werden Restmaterialien durch blaue Farbe graphisch dargestellt. Bei der Oberflächenextraktion werden diese Restmaterialinformationen, die im Stixelpunkt des simulierten Werkstücks gespeichert werden, weiter von den erzeugten Dreiecken vererbt. Das Dreieck, welches den Restmaterial-Stixelpunkt enthält, wird mit blauer Farbe gezeigt.

6.4 Zusammenfassung

Die in Kapitel 6 vorgestellten Methoden zur Erkennung der Materialverletzungen und Ermittlung der Restmaterialien sollen eine Optimierung eines fünffachsignen NC-Programms ermöglichen.

Die voxelbasierte Vorgehensweise zur Überprüfung der Qualität des simulierten Werkstücks zeigt deutlich, dass diese Methode eine Überprüfung des Werkstücks in der NC-Simulation zulässt und vereinfacht. Statt der Ermittlung der Abweichungen zwischen den Oberflächen des simulierten Werkstücks und den Soll-Flächen werden hierbei nur die Abstände der Stixelpunkte des simulierten Werkstücks zu den Soll-Flächen berechnet. Die erkannten Materialverletzungen und Restmaterialien sind sowohl in dem simulierten Werkstückmodell gespeichert als auch graphisch auf dem simulierten Werkstück dargestellt. Die Informationen über die fehlerhaften NC-Zeilen werden in den entsprechenden Programmprotokollen aufgezeichnet.

Der Vorteil dieser graphischen Darstellung liegt in der Einfachheit und Universalität der Methoden. Die erkannten Materialverletzungen und Restmaterialien sind durch farbige Dreiecke darstellbar. Damit kann der Programmierer eine Überprüfung des NC-Programmes vornehmen. Ein weiterer Vorteil dieses Verfahrens besteht in der einfachen Berechnung der Materialverletzungen und der Restmaterialien. Dies erlaubt wiederum eine detaillierte Korrektur des fehlerhaften NC-Programmes.

Eine einheitliche Lösung für eine NC-Programmkorrektur für das fünfachsiges Fräsen ist nicht möglich. Die verschiedenen Bearbeitungsfälle einerseits und die verschiedenen Fräserarten andererseits erfordern eine differenzierte Betrachtung der verschiedenen Fälle. Eine Korrektur des NC-Programmes ist dann sinnvoll, wenn die betroffenen Geometrien des Werkstücks im gegebenen Fall genau analysiert werden. Der Einsatz eines CAD-Systems zur Untersuchung der Eigenschaften der Werkstückgeometrie ist erforderlich. Eine detaillierte Korrektur des fehlerhaften NC-Programms ist in dem zukünftigen Korrektursystem vorgesehen.

7 Systemrealisierung und Anwendungsbeispiel

7.1 Funktionsbeschreibung

In diesem Kapitel werden zuerst der Aufbau und die Realisierung eines voxelbasierten fünfachsigen NC-Simulationssystems vorgestellt. Danach wird ein Beispiel zur fünfachsigen Simulation des NC-FräSENS eines Verdichterlaufrades gezeigt.

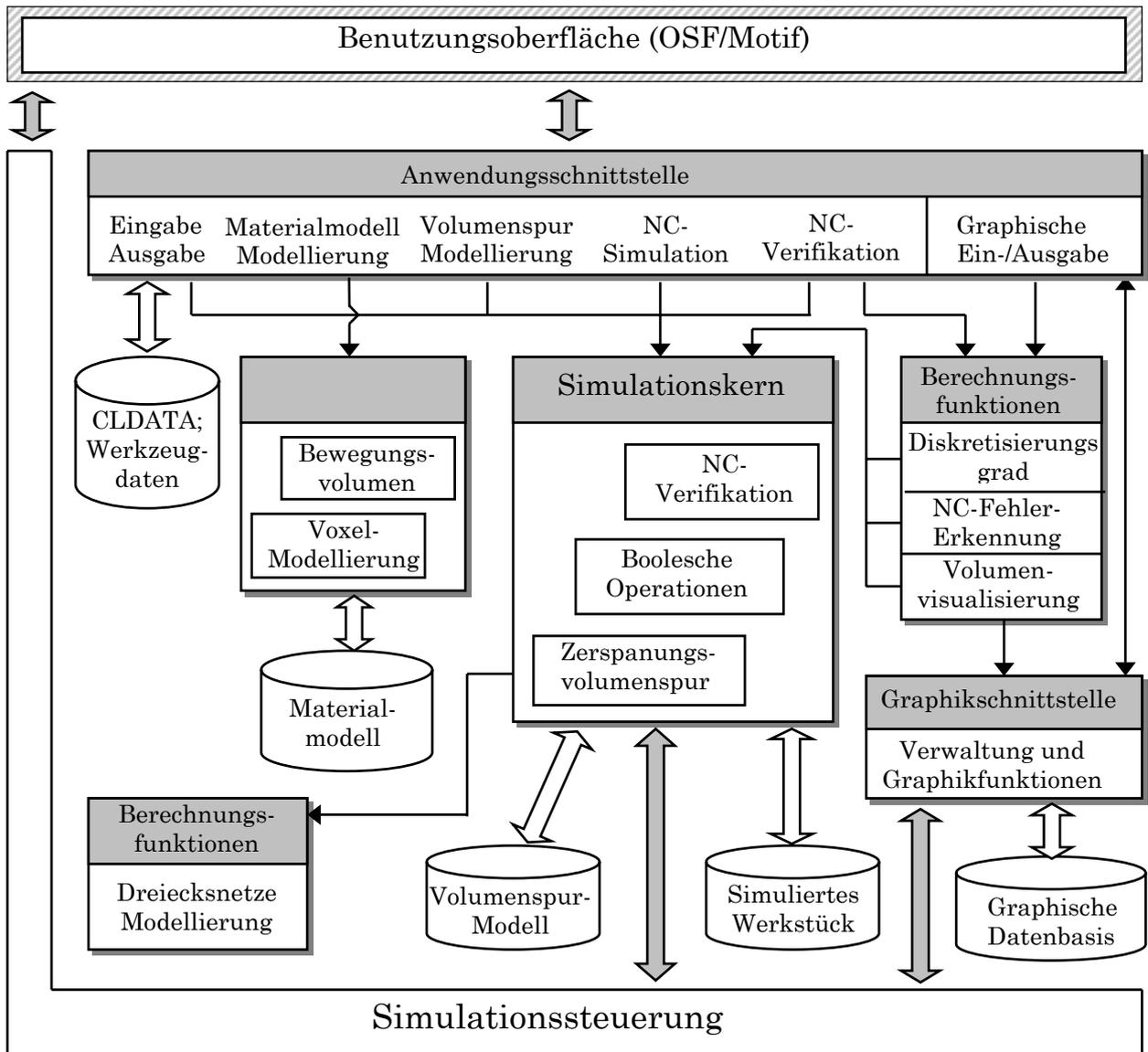


Bild 7.1: Modularer Systemaufbau

Ein wesentliches Ziel der Arbeit ist es, ein NC-Simulationssystem für einen breiten Anwendungsbereich zu entwickeln und damit für möglichst viele interessierte Anwender verfügbar zu machen. Im

Hinblick auf die rasante Entwicklung der Rechnertechnologie ist die Systemstruktur funktionsorientiert modular aufgebaut, so dass die Übertragbarkeit des funktionalen Konzepts auf andere Systeme (Mikroprozessoren) und Parallelrechner gewährleistet ist. Bild 7.1 stellt die Module und deren Schnittstellen des realisierten voxelbasierten fünfachsigen NC-Simulationssystems dar. Die Module werden in diesem Kapitel noch näher beschrieben.

Die Software des NC-Simulationssystems wurde in der rechnerunabhängigen Programmiersprache C++ [111] geschrieben und auf einer Workstation SGI Onyx mit vier 150 MHz MIPS R4400 CPU unter dem Betriebssystem IRIX realisiert. Die durchgeführten Berechnungen basieren auf Gleitkommazahlen mit doppelter Genauigkeit.

Mit dem nachfolgend beschriebenen NC-Simulationssystem wird die reale NC-Bearbeitung des Werkstücks simuliert und auf mögliche Materialverletzungen und Restmaterialien untersucht.

Analog zur realen NC-Bearbeitung wird das Materialmodell zuerst nach den maximalen Maßen des zu bearbeitenden Werkstücks modelliert. Diese Aufgabe wird von einem VCM-Modul, welches am Institut für Produktionsanlagen und Konstruktionstechnik Berlin entwickelt wurde und als ausführbares Modul von dem Simulationssystem bei Bedarf geladen werden kann, wahrgenommen. Die vom Konstrukteur vorgegebene NC-Simulationstoleranz zur Kontrolle der Simulationsfehler wird in diesem Modul berücksichtigt. Das erzeugte voxelbasierte Materialmodell wird in der Systemdatenstruktur abgelegt und lässt sich vom Simulationskern aufrufen.

Der Hauptteil des voxelbasierten fünfachsigen NC-Simulationssystems ist der Simulationskern, der über eine prozedurale Schnittstelle grundlegende Funktionen zur Werkstückaktualisierung anbietet. Entsprechend dem Ansatz ist der Simulationskern aus Komponenten aufgebaut, welche getrennte Funktionalitäten zur Berechnung einerseits des Zerspanungsvolumens des Werkzeugs und andererseits der Booleschen Operationen und der NC-Verifikation gewährleisten. Zur Vereinfachung und Beschleunigung der NC-Simulation werden die Booleschen Operationen impliziert durchgeführt. Bei der NC-Verifikation wird eine Voxel-Struktur zur Lokalisierung der Soll-Flächen und zur Auswahl eines anfänglichen Flächenpunkts für die Berechnung der Abweichung zwischen den Oberflächen des simulierten Werkstücks und den Soll-Flächen eingesetzt.

Für die Berechnung des voxelbasierten Zerspanungsvolumens und der Verifikation des NC-Programms wurden C++-Funktionen des Geometrie-Modellierkerns (ACIS⁵) der Firma Spatial

⁵ ACIS ist ein Geometrie Modellierkern der Firma Spatial Technologies.

Technologies eingesetzt [112]. Ebenso wurde das voxelbasierte fünfsichtige NC-Simulationssystem in der objektorientierte Programmiersprache C++ realisiert. Hieraus ergaben sich in der Entwicklung Vorteile hinsichtlich der klaren Unterscheidbarkeit von Anwendungs- und Kernfunktionen, die als Methoden der umgesetzten Objektklassen bereitgestellt werden, sowie hinsichtlich der übersichtlichen Aggregation der Datenelemente und ihrer verarbeitenden Methoden.

Gesonderte gebundene Module sind die Berechnungsfunktionen des Marching-Cubes-Verfahrens zur Volumenvisualisierung und die Berechnungsfunktionen zur Erkennung von Materialverletzungen und Restmaterialien. Beide Verfahren bauen Datenfelder und Tabellen als Zwischenspeicher für eine echtzeitnahe Visualisierung auf. Auslöser der Aktualisierung sind im Fall der Volumenvisualisierung die NC-Programme und im Fall der Herausfindung der Materialverletzungen die Fehlerdarstellung.

Die Verwaltung der benötigten graphischen Datenbasis erfolgt über die graphische Schnittstelle Open Inventor⁶ [113, 114, 115, 116]. Kennzeichnend für diese Graphikschnittstelle ist die flexible Konfigurierbarkeit hierarchischer Graphikstrukturen bei gleichzeitiger hoher Aufzeichnungsgeschwindigkeit. Besonders vorteilhaft ist die Eigenschaft zum Aufbau der segmentierten graphischen Datenbasis, welche unmittelbar eine lokal eingegrenzte wiederholte Aktualisierung des Volumens in definierten Bildaufbauarten erlaubt.

Die bei der NC-Verifikation herausgefundenen Materialverletzungen und Restmaterialien werden auf der Oberfläche des simulierten Werkstücks nicht nur graphisch dargestellt, sondern auch zusammen mit entsprechenden Flächeninformationen in einem Programmprotokoll gespeichert. Dieses wird für eine weitere Optimierung der NC-Programme zur Verfügung gestellt.

7.2 Benutzungsoberfläche

7.2.1 Eingesetztes Graphiksystem

Die Benutzungsoberfläche dient der Präsentation der graphischen Darstellung des Bearbeitungsstandes während der Simulation und der Interaktion zwischen Benutzer und System aufgrund wechselseitig austauschbarer Informationen[117].

⁶ Open Inventor ist ein eingetragenes Warenzeichen der Firma Silicon Graphics Inc.

Zur Entwicklung der graphischen Benutzungsoberfläche wurde in der vorliegenden Arbeit eine „Schnittstellen-Werkzeugkasten-RapidApp“, die von der Firma Silicon Graphics Inc. entwickelt wurde, eingesetzt. Sie stellt dem Anwender die gebräuchlichsten Elemente von Benutzungsoberflächen zur Verfügung. Dazu gehören graphische Bedienelemente, wie Buttons, Menüs und Scrollbars usw., welche für bestimmte Interaktionsaufgaben konzipiert sind. Durch einfaches Zusammenstellen dieser vorgefertigten Elemente lassen sich komplexe Benutzungsoberflächen erstellen. Die Verwendung des Schnittstellen-Werkzeugkasten-RapidApp bietet nicht nur den Vorteil, Benutzungsoberflächen schneller entwerfen zu können, sondern gewährleistet auch, dass Oberflächen verschiedener Applikationen einheitlich aussehen und zu bedienen sind [118].

Die von RapidApp generierten Programme sind aus objektorientierten C++-Programmcodes und weiteren Schnittstellen-Klassen, die auf IRIS ViewKit basieren, zusammengesetzt. IRIS ViewKit bietet eine leistungsstarke Benutzungs- und Anwendungsumgebung. Mittels vordefinierter Schnittstellen ermöglicht IRIS ViewKit dem Anwender eine schnelle und einfache Integration der graphischen Schnittstelle Open Inventor und OpenGL.

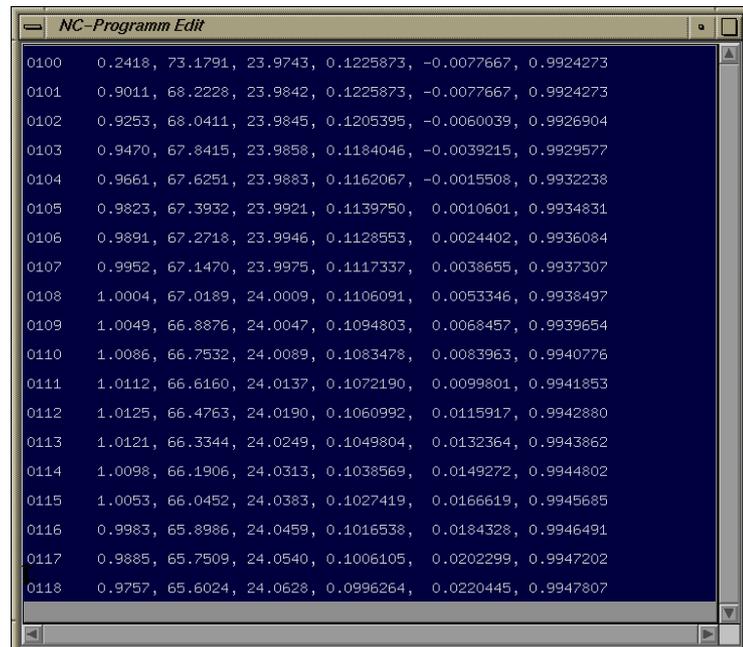
Die graphische Schnittstelle Open Inventor, welche ebenfalls von der Firma Silicon Graphics Inc. entwickelt wurde, ist die Erweiterung der OpenGL. Die OpenGL ist ein hardwareunabhängiges Werkzeug zur Entwicklung von Graphikprogrammen. Die graphische Schnittstelle Open Inventor beschreibt eine Programmierschnittstelle zur Definition, Modifikation, Interaktion und Darstellung hierarchisch gegliederter, graphischer Daten.

Die mächtigen graphischen Schnittstellen (Open Inventor und Open GL) stellen alle Funktionen zur Verfügung, die heute von graphischen Anwendungen gefordert werden [113, 115, 116, 119]. Dies gilt insbesondere für die dynamische Manipulation komplexer Strukturen, wie die bei der dynamischen 3D-Simulation. Die hierarchische Strukturierung graphischer Objekte ermöglicht eine schnelle und einfache Definition (Modifikation) von Bildinhalten. Sie erleichtert das Auswählen und Identifizieren von Teilobjekten. Durch mehrfache Verwendung graphischer Objekte kann der Speicherbedarf auch minimiert werden.

7.2.2 Der realisierte Benutzerdialog

Eine verwendete Benutzungsoberfläche sorgt für eine einheitliche Interaktionssteuerung und übernimmt Funktionen wie die Verwaltung von Fenstern, Menüs, graphischen Objekten, die Ausgabe von Objekten und Meldungen sowie die Steuerung der Dialogfolge. Sie benutzt direkte Funktionen des Graphiksystems.

Mit Hilfe der Benutzungsoberfläche werden das zu simulierende NC-Programm und die Maße des Werkstückmodells zur Berechnung der Größe des Materialmodells eingegeben. Die Graphikparameter für die Ausgabe werden eingestellt und der Simulationslauf wird gestartet. Der Simulationsablauf unterscheidet sich in zwei Betriebsarten, nämlich automatischer Simulation und manueller Simulation. Bei der automatischen Simulation wird das ganze NC-Programm in der Systemstruktur aufgerufen und dann abgearbeitet. Bei der manuellen Simulation wird das NC-Programm durch ein sogenanntes „Programm-Edit“ (Bild 7.2) kontrolliert abgearbeitet. Bild 7.3 zeigt den Dialogbaum des Simulationssystems.



Line	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6	Value 7
0100	0.2418	73.1791	23.9743	0.1225873	-0.0077667	0.9924273	
0101	0.9011	68.2228	23.9842	0.1225873	-0.0077667	0.9924273	
0102	0.9253	68.0411	23.9845	0.1205395	-0.0060039	0.9926904	
0103	0.9470	67.8415	23.9858	0.1184046	-0.0039215	0.9929577	
0104	0.9661	67.6251	23.9883	0.1162067	-0.0015508	0.9932238	
0105	0.9823	67.3932	23.9921	0.1139750	0.0010601	0.9934831	
0106	0.9891	67.2718	23.9946	0.1128553	0.0024402	0.9936084	
0107	0.9952	67.1470	23.9975	0.1117337	0.0038655	0.9937307	
0108	1.0004	67.0189	24.0009	0.1106091	0.0053346	0.9938497	
0109	1.0049	66.8876	24.0047	0.1094803	0.0068457	0.9939654	
0110	1.0086	66.7532	24.0089	0.1083478	0.0083963	0.9940776	
0111	1.0112	66.6160	24.0137	0.1072190	0.0099801	0.9941853	
0112	1.0125	66.4763	24.0190	0.1060992	0.0115917	0.9942880	
0113	1.0121	66.3344	24.0249	0.1049804	0.0132364	0.9943862	
0114	1.0098	66.1906	24.0313	0.1038569	0.0149272	0.9944802	
0115	1.0053	66.0452	24.0383	0.1027419	0.0166619	0.9945685	
0116	0.9983	65.8986	24.0459	0.1016538	0.0184328	0.9946491	
0117	0.9885	65.7509	24.0540	0.1006105	0.0202299	0.9947202	
0118	0.9757	65.6024	24.0628	0.0996264	0.0220445	0.9947807	

Bild 7.2: Editiertes NC-Programm

Am Anfang werden die Soll-Maße des Werkstücks, CLDATA, Parameter des Werkzeugs und die Simulationstoleranz eingegeben. Von den Soll-Maßen des Werkstücks wird die Größe des Materialmodells abgeleitet und anschließend bei der Modellierung des Materialmodells als Ausgangsmaß benutzt, wobei der Diskretisierungsgrad des Voxelmodells von der Simulationstoleranz bestimmt wird. Das erzeugte Materialmodell dient als ein Basismodell für die NC-Simulation. Bild 3.13 stellt ein solches Materialmodell dar.

Bei der manuellen Simulation kann der Benutzer zuerst in einem Programm-Edit, das mittels manueller NC-Simulation aktiviert wird, sowohl die Zielposition der NC-Zeile auswählen als auch beliebig eingeben. Hierbei werden die Materialverletzungen mit der Betriebsart der simulationsbeglei-

tenden Prüfung untersucht. Jedoch findet die Ermittlung der Restmaterialien hier nicht statt, da das ganze Werkstück in diesem Fall noch nicht fertiggestellt wird. Nach jedem Bearbeitungsschritt wird das simulierte Werkstück mit den Materialverletzungen, falls diese ermittelt wurden, graphisch dargestellt.

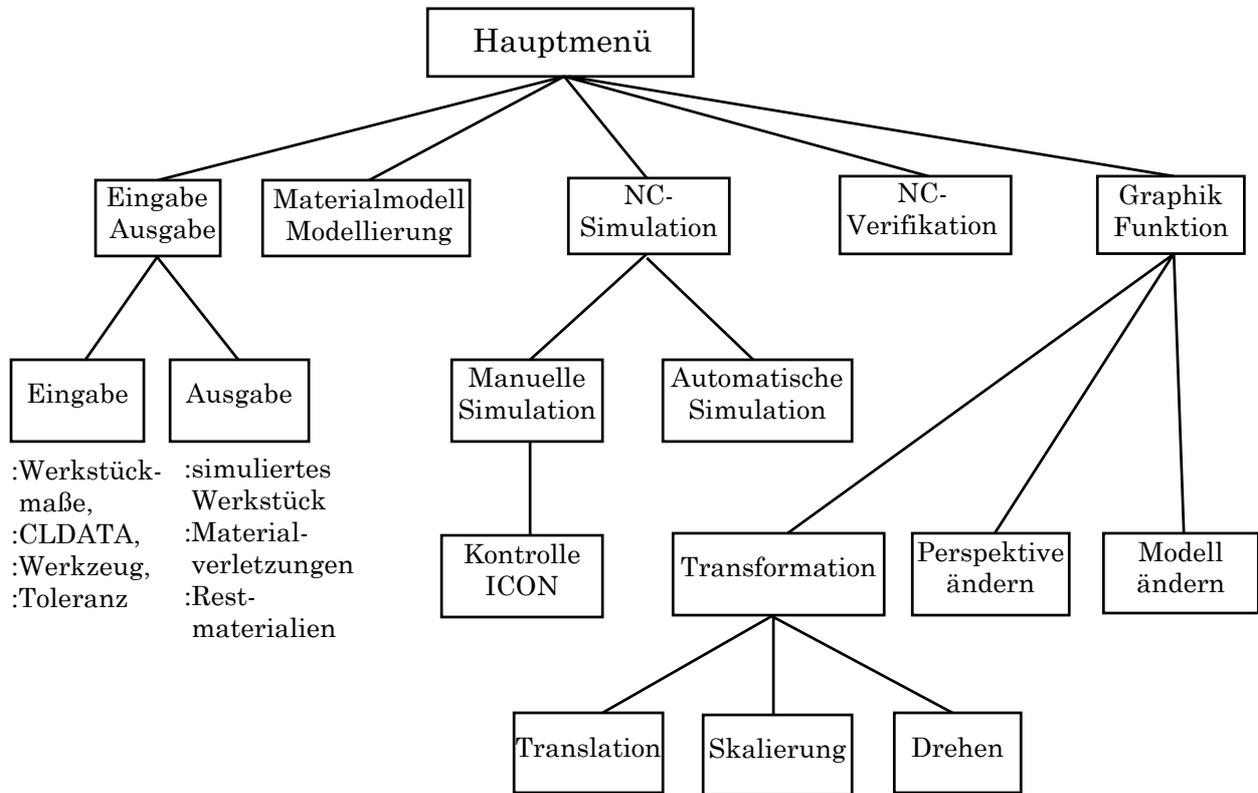


Bild 7.3: Dialogbaum

Im Gegensatz dazu wird bei der automatischen Simulation das vollständige eingegebene NC-Programm ausgewählt. Danach kann die Simulation gestartet werden, wobei sämtliche Bearbeitungsschritte ohne Überprüfung der Materialverletzungen und Restmaterialien abgearbeitet werden. Die Erkennung der Materialverletzungen und die Ermittlung der Restmaterialien finden in diesem Fall erst nach Beendigung der NC-Simulation durch Aufrufen der Funktion NC-Verifikation statt. In dieser Situation werden die Materialverletzungen durch Anwendung des Batch-Betriebs untersucht. Die NC-Simulation wird dadurch hervorragend beschleunigt, da die zeitaufwendige Überprüfung der Materialverletzungen und Restmaterialien nicht in der NC-Simulation durchgeführt wird.

7.2.3 Graphische Darstellungsmöglichkeiten

Bei der graphischen Darstellung des simulierten Werkstücks kann der Benutzer unter folgenden Darstellungsarten eine Auswahl treffen:

- Oberflächenmodell,
- Voxelmodell und
- Ausblenden.

Die Auswahl dafür wird mit dem Pop-Up-Menü ‚Graphische Darstellung‘ durchgeführt. Bei der Darstellung des Oberflächenmodells müssen die Oberflächen des simulierten Werkstücks zuerst extrahiert werden. Der Extraktionsprozess belastet aber die Simulationsgeschwindigkeit durch zusätzliche Rechenleistung, die viel Zeit in Anspruch nehmen kann. Bei der Voxelmodell-Darstellung dagegen wird die Oberflächenextraktion hinfällig, da die Visualisierung des voxelbasierten Werkstücks direkt mit der Graphikbibliothek Open Inventor, die auf die GL Graphikbibliothek aufsetzt und in hohem Maße auf die Graphik-Hardware zugeschnitten ist, erfolgt. Das hat eine starke Beschleunigung des Simulationsprozesses zur Folge. Jedoch verschlechtert sich in diesem Fall die Oberflächenqualität des simulierten Werkstücks (Bild 5.5). Mit ‚Ausblenden‘ wird die graphische Darstellung im NC-Simulationsprozess ausgeschaltet. Der direkte Vorteil ist die drastische Beschleunigung des NC-Simulationsprozesses. Diese Situation entspricht der Verifikationsart ‚Batch-Betrieb‘. Am Schluss der NC-Simulation kann die Oberfläche des simulierten Werkstücks in realitätsnaher Darstellung nachträglich gezeigt werden.

Neben den anwendungsabhängigen graphischen Funktionen kommen auch die permanent aktiven graphischen Funktionen zur Anwendung. Die permanent aktiven graphischen Funktionen dienen beispielsweise der Manipulation der Graphikobjekte und sind zu jeder Zeit für die Anwendung verfügbar.

Hierzu zählen:

- Translation,
- Rotation,
- Skalierung,
- Orthogonale Parallelprojektion,

- Perspektive Projektion,
- Farbschattieren,
- Ein- und Ausblenden der schattierten Darstellung und
- Ein- und Ausblenden der Dreiecksnetzdarstellung.

7.3 Simulationssteuerung

Dieses Modul dient zum Koordinieren der Abläufe während der NC-Simulationsbearbeitung und stellt dafür Funktionen wie Initialisierung und Ausführung zur Verfügung. Diese Funktionen werden von der Benutzungsoberfläche angesprochen, die wiederum Funktionen des NC-Simulationskerns aktiviert. Danach gibt der Simulationskern die Kontrolle wieder an die Benutzungsoberfläche mit einer Meldung und einer graphischen Darstellung über das Simulationsergebnis zurück.

Es wurden zwei verschiedene Simulationsarten, manuelle und automatische Simulation, entwickelt. Bei der manuellen Simulation werden die gewünschten Bewegungsinformationen, welche die Anfangs- und Zielposition enthalten, direkt von der Benutzungsoberfläche mitgeliefert. Die Ermittlung von Materialverletzungen kann nach der Auswahl des Programmierers in der NC-Simulation begleitend durchgeführt oder nicht durchgeführt werden. In Bild 7.4 wird der Simulationsablauf bei der manuellen NC-Simulation gezeigt.

Im Gegensatz dazu handelt es sich bei der automatischen Simulation um die Simulation eines NC-Programmes. Nach Auswahl des NC-Programmes kann die Simulation durchgeführt werden. Der Unterschied zu der manuellen NC-Simulation ist, dass das vollständige NC-Programm automatisch abgearbeitet und in der NC-Simulation keine Prüfung der Materialverletzungen durchgeführt wird. Die Erkennung der Materialverletzungen und die Ermittlung der Restmaterialien finden erst nach Ablauf der NC-Simulation durch die NC-Verifikation statt. Bild 7.5 stellt den NC-Simulationsablauf der automatischen NC-Simulation dar.

In beiden Situationen kann der Simulationsablauf jederzeit unterbrochen werden, wobei der aktuelle NC-Satz vollständig abgearbeitet wird, um einen definierten Zustand zu erreichen.

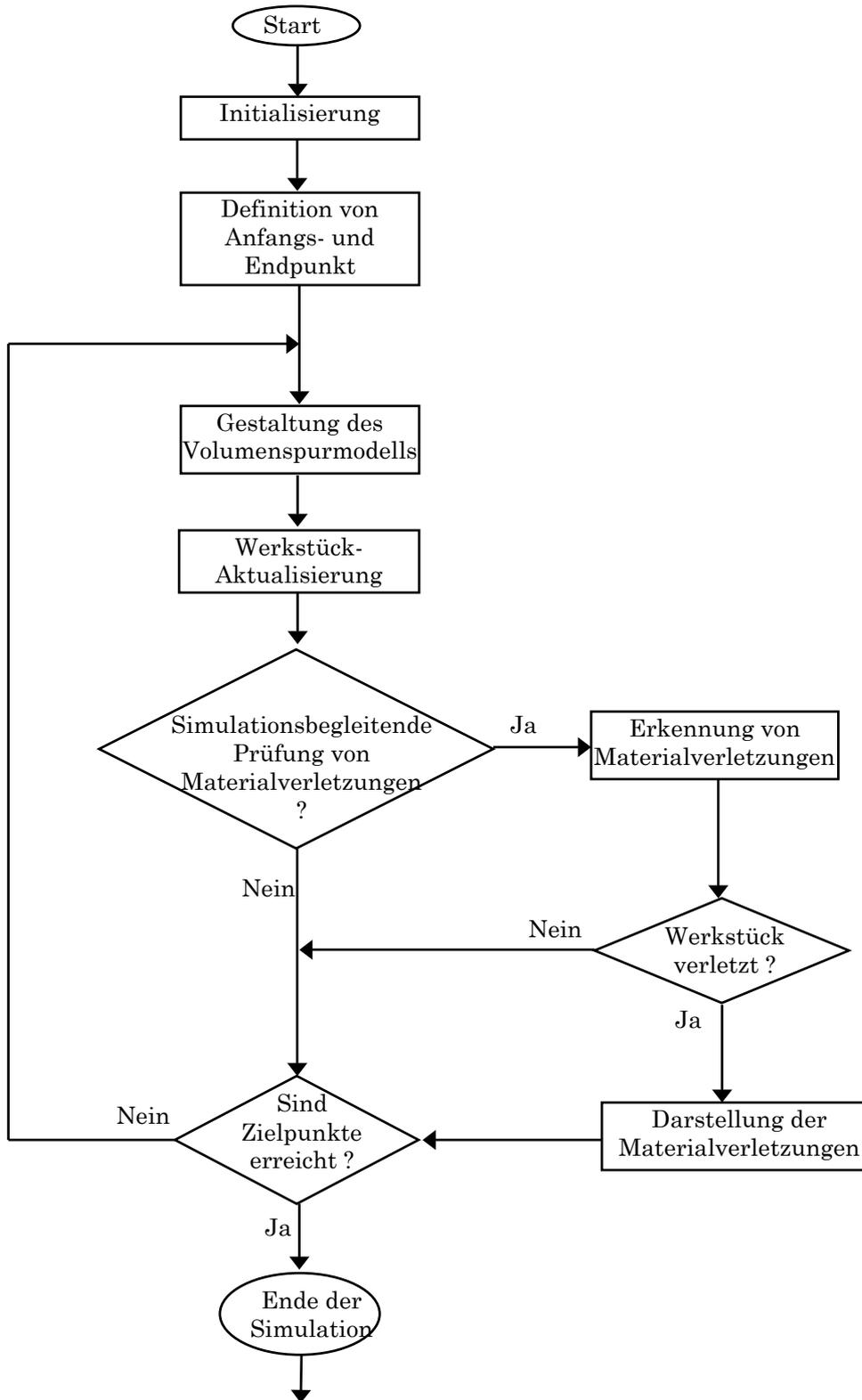


Bild 7.4: Simulationsablauf bei der manuellen NC-Simulation mit der begleitenden Prüfung der Materialverletzungen

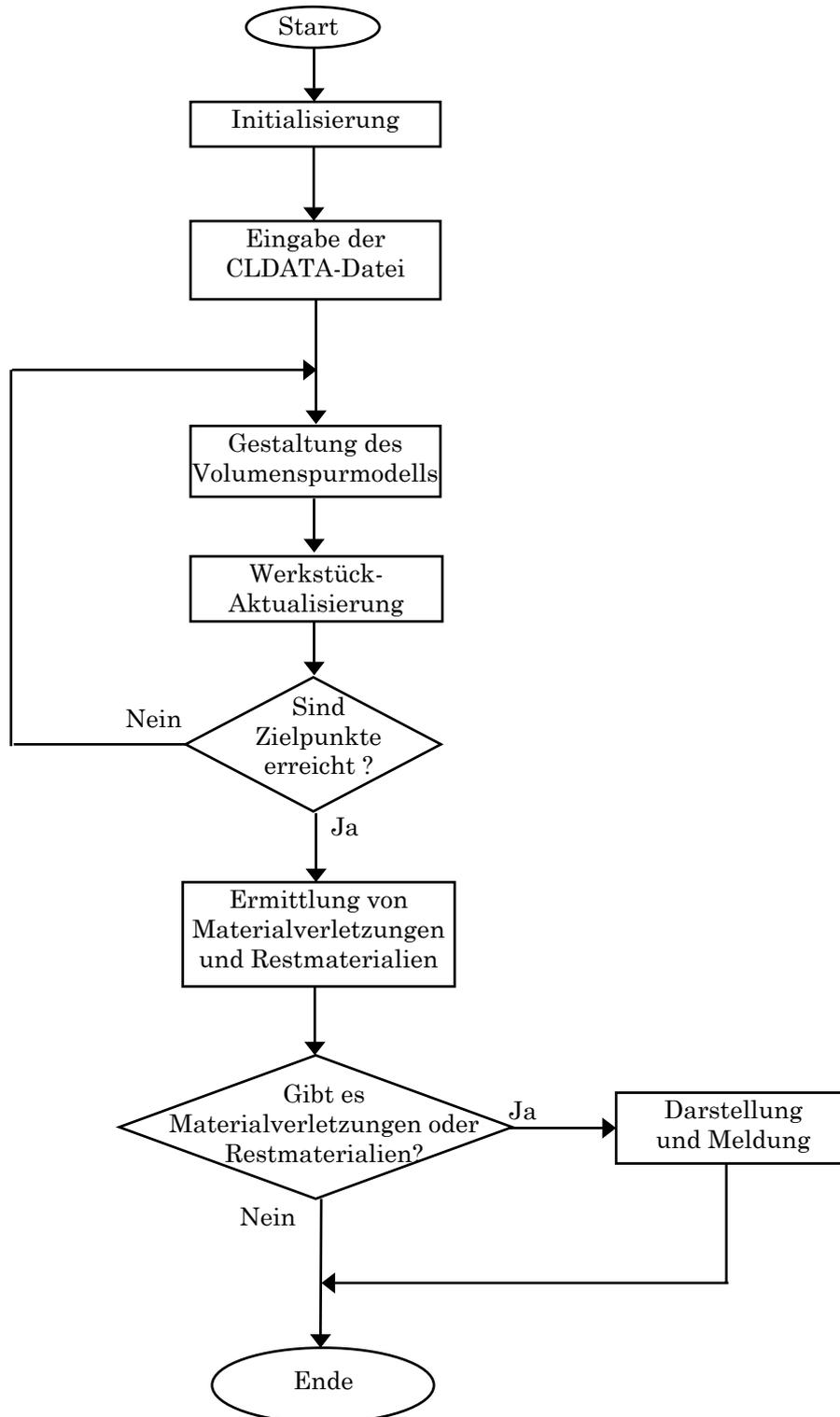


Bild 7.5: Simulationsablauf der automatischen NC-Simulation mit der NC-Verifikation (Batch-Betrieb)

7.4 Anwendungsbeispiel

7.4.1 Beschreibung eines Verdichterlaufrades

In diesem Unterkapitel wird ein Anwendungsbeispiel zur fünfachsigigen NC-Simulation gegeben. Als ein aussagekräftiges Muster wurde ein Verdichterlaufrad ausgewählt.

Die wesentlichen Bestandteile eines Gasturboladers sind zwei Laufräder. Die Leistung und Lebensdauer eines Gasturboladers werden direkt von der Qualität der Laufräder beeinflusst. Die Oberflächen eines Laufrades werden gewöhnlich durch 3D-geformte Flächen wie Regelflächen und Rotationsflächen modelliert. Um die Qualität von Laufrädern zu erhöhen, wird heutzutage vorwiegend das fünfachsigige NC-Bearbeitungsverfahren zu deren Herstellung eingesetzt. Bild 7.6 zeigt beispielsweise einen Abgasturbolader mit zwei Laufrädern[120].

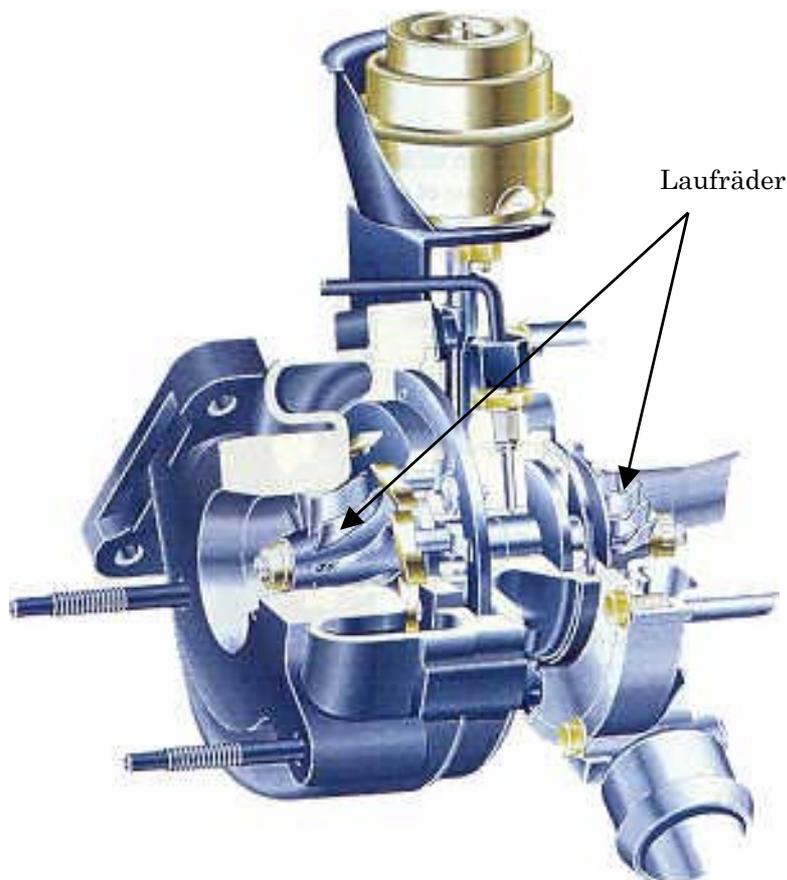


Bild 7.6: Abgasturbolader

In der vorliegenden Arbeit wurde ein Verdichterlaufrad als Beispiel für die voxelbasierte fünfachsigige Simulation des NC-FräSENS ausgewählt. Dazu wurde ein NC-Programm für die fünfachsigige NC-FräSBearbeitung dieses Verdichterlaufrades entwickelt. Daraufhin erfolgte die NC-Simulation zur Überprüfung und Optimierung des NC-Programms. Bild 7.7 stellt eine Zeichnung des Verdichterlaufrades, das mit AutoCAD erzeugt wurde, dar. Die Maßeinheit ist mm.

Das Verdichterlaufrad hat 12 Schaufeln, die mittels Regelflächen gestaltet werden. Die Radnabe wird durch eine Rotationsfläche gebildet.

Nach den Erfahrungen aus der Praxis sollten die Schaufeln mittels dem Umfangfräs-Verfahren und die Radnabe mittels dem Stirnfräs-Verfahren bearbeitet werden. Die Materialverletzungen für das vollständige Verdichterlaufrad sollten 0.1 mm nicht überschreiten. Die Restmaterialtoleranz für die Schaufelflächen war mit 0.1mm und für die Radnabe mit 0.5 vorgegeben. Eine gleichmäßige NC-Rillenverteilung auf der Radnabe war eine weitere Anforderung. Nach den Erfahrungen von Konstrukteuren tragen die gleichmäßig verteilten NC-Rillen auf der Radnabe zur Erhöhung der Strömungsqualität des Verdichterlaufrades bei.

Entsprechend den Maßen des Verdichterlaufrades wurde die Größe des Materialmodells für die NC-Simulation abgeleitet.

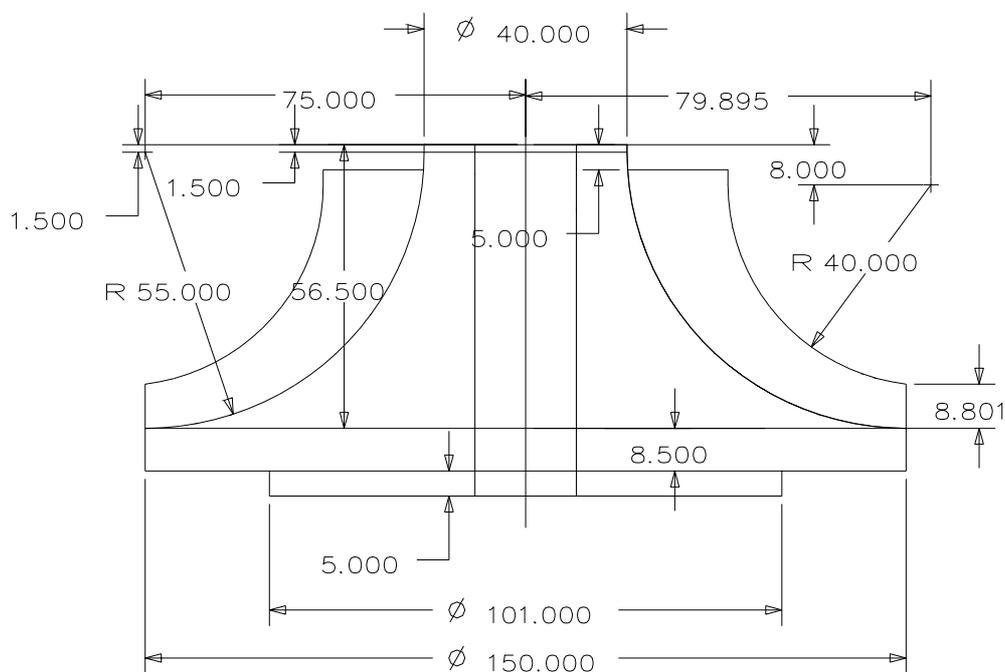


Bild 7.7: Zeichnung eines Verdichterlaufrades (AutoCAD)

7.4.2 NC-Programme eines Verdichterlaufrades

Die NC-Programmierung⁷ wurde mit dem im Nationalen Speziellen CAD/CAM-Forschungszentrum entwickelten NPU-CAD/CAM-System⁸ durchgeführt. Das System besteht aus folgenden Subsystemen:

- NPU-GDS: Graphical Display System,
- NPU-CCSFIT: Complex Curve und Surface Fitting-System,
- NPU-GNCP/SS: Graphical Numerical Control Programming/Sculptural Surface und
- NPU-NCPPost: NC Post Processorsystem.

Bei der NC-Programmierung wurden zuerst mit dem NPU-CCSFIT-System eine Schaufelfläche mit abwickelbarer Regelfläche und eine Radnabefläche mit Rotationsfläche modelliert. Die NC-Programmierung erfolgte in zwei Schritten mittels NPU-GNCP/SS:

- NC-Programmierung einer Schaufelfläche und
- NC-Programmierung der Radnabefläche.

Die Schaufelfläche wurde vorzugsweise durch Umfangsfräsen mit einem zylindrischen Gesenkfräser (Radius 4.0 mm) unter Berücksichtigung von Grenzflächen erzeugt. Die Grenzflächen für die Schaufelflächenbearbeitung bestehen aus zwei Schaufelflächen und der Radnabefläche. Diese zwei Schaufelflächen wurden durch Rotation der ursprünglichen Schaufelfläche hergestellt.

Die Radnabefläche wurde durch Stirnfräsen mit einem zylindrischen Gesenkfräser (Radius 4.0 mm) unter Berücksichtigung von Grenzflächen gefertigt. Die Grenzflächen für die Radnabebearbeitung bestehen aus zwei Schaufelflächen.

Die NC-Programmierung erfolgte unter Berücksichtigung der vom Konstrukteur vorgegebenen Toleranzen. Es wurden zwei NC-Programme erzeugt. Bild 7.8 stellt die Fräserbahnen für Schaufel und Radnabe dar.

⁷ Bei der NC-Programmierung wurde der Verfasser von den Mitarbeitern des Nationalen Speziellen CAD/CAM-Forschungszentrums an NPU in Xi'an unterstützt.

⁸ NPU ist die Abkürzung für die „Northwestern Polytechnical University in China.“

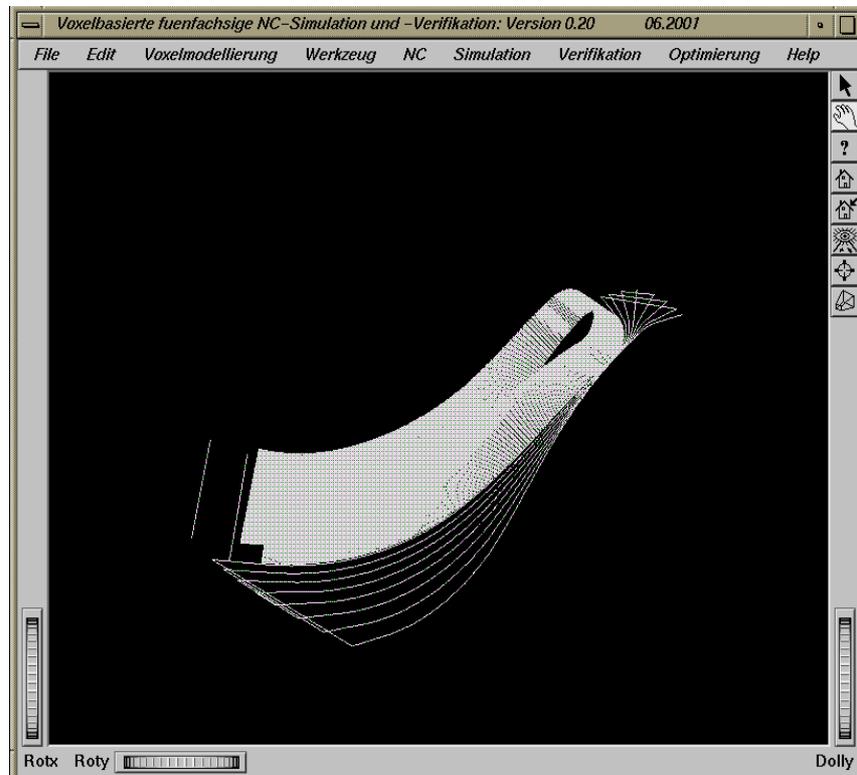
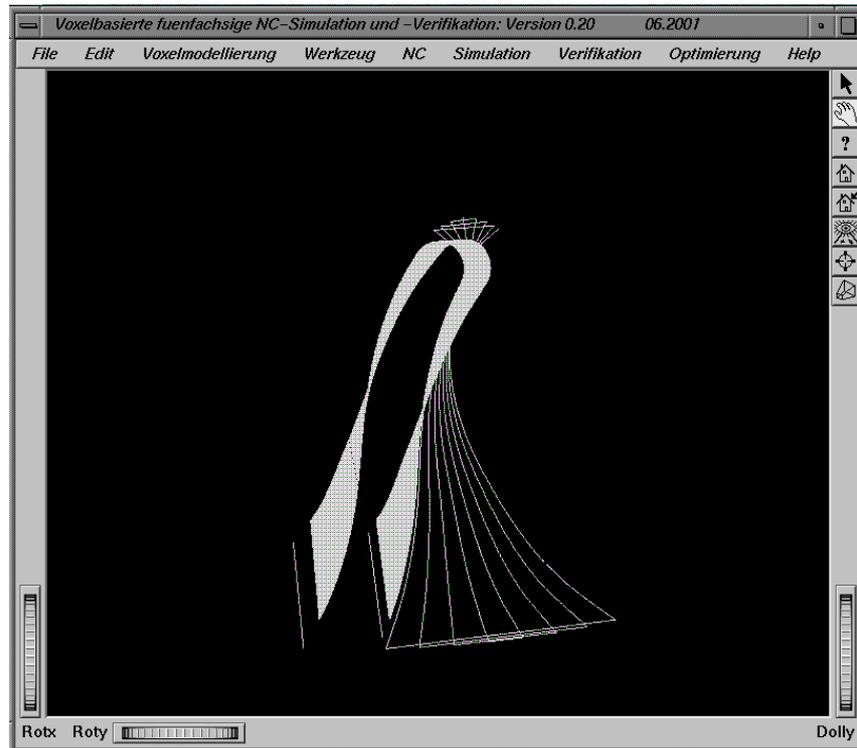


Bild 7.8: Darstellung der Fräserbahnen für Schaufel und Radnabe eines Verdichterlaufrades

7.4.3 Fünffachsiges NC-Simulation eines Verdichterlaufrades

Die NC-Simulation beginnt mit der Initialisierung des Systems. Dabei werden Parameter, wie lokale und globale Systemstrukturdaten initialisiert. Danach folgt die Eingabe von Werkzeugparametern, vorgegebenen Simulationstoleranzen, Programmierungstoleranzen und den maximalen Abmessungen des Verdichterlaufrades (siehe 7.4.1). Mit dem Fräserradius und den vorgegebenen Simulationstoleranzen wird der Diskretisierungsgrad für das Materialmodell des Verdichterlaufrades berechnet. Durch die maximalen Abmessungen wird die Größe des Materialmodells des Verdichterlaufrades bestimmt. Die Modellierung des Materialmodells des Verdichterlaufrades für die NC-Simulation erfolgt durch das VCM-System.

In diesem Beispiel wurde ein zylindrischer Gesenkfräser mit dem Radius 4.0 mm eingesetzt. Ausgehend von den vorgegebenen Programmierungstoleranzen für Materialverletzungen (0.1 mm für das vollständige Verdichterlaufrad) und Restmaterialien (0.5 mm auf der Radnabe) wurde die Simulationstoleranz (0.05 mm) bestimmt. Nach der im Abschnitt 3.4 beschriebenen Methode wurde der Diskretisierungsgrad (Voxelgröße 0.5 mm x 0.5 mm x 0.5 mm, siehe Tabelle 3.3) für das Voxelmodell ausgewählt. Für das VCM-System wurde ein Würfel mit den Größen 160.0 mm x 160.0 mm x 65.0 mm als Materialmodell definiert (siehe 3.5).

Im Allgemeinen werden bei der NC-Programmierung nur zwei NC-Programme (für Schaufelfläche und Radnabefläche) für die NC-Fräsbearbeitung eines Verdichterlaufrades hergestellt (siehe 7.4.2). In der Praxis jedoch wird das Werkstückmaterialmodell jeweils um 30° um die Z-Achse gedreht, um damit sämtliche 12 Schaufeln und 12 Teilflächen der Radnabe bearbeiten zu können. Analog zur NC-Bearbeitung wurden in der NC-Simulation jedoch die NC-Programme jeweils um 30° um die Z-Achse rotiert. Daraufhin wurden 12 NC-Programme für Schaufeln und 12 NC-Programme für die Teilflächen der Radnabe erzeugt. Bild 7.9 stellt die Fräserbahnen zur NC-Bearbeitung des vollständigen Verdichterlaufrades dar.

Das System arbeitet satzweise alle vorgegebenen NC-Sätze der NC-Programme ab. Die NC-Verifikation wurde mit dem ‚Batch-Betrieb‘-Verfahren durchgeführt. Nach der NC-Simulation wurden die möglichen Materialverletzungen und die Restmaterialien ermittelt. Die Ergebnisse der Überprüfung haben gezeigt, dass die NC-Programme für das Verdichterlaufrad den Anforderungen an die Produktqualität vollständig gerecht werden. Die Materialverletzungen wurden unter 0.1 mm und die Restmaterialien auf der Radnabe unter 0.5 mm gehalten. Bild 7.10 stellt das simulierte Verdichterlaufrad dar. Es ist ersichtlich, dass die NC-Rillen auf der Radnabe gleichmäßig verteilt sind.

Der Anwender ist auch in der Lage, das simulierte Verdichterlaufrad von jeder Richtung und jedem Standpunkt aus im 3D-Raum zu beobachten.

Für die Zeitmessung wurde die Zeitfunktion, die in der Funktionsbibliothek der Programmiersprache ‚C++‘ zur Verfügung gestellt wird, herangezogen. Diese Zeitfunktion ‚clock‘ liefert Zeitinformationen in 10 ms-Auflösung. Sie wird vor und nach der Ausführung der Simulationsfunktionen aufgerufen. Die Differenz zwischen diesen Zeiten bildet die Ausführungszeit für das Bewegungsintervall (entsprechend einem NC-Satz) des Fräsens. Die Laufzeit des NC-Programms ergibt sich aus der Summe aller Ausführungszeiten für die gesamten NC-Sätze. Während der Zeitbetrachtung werden die Minimum- und Maximum-Werte für jede Simulationsfunktion ermittelt. Es ergab sich folgendes Ergebnis: Für das vollständige Verdichterlaufrad wurden 591 Minuten benötigt, für eine Schaufel und eine Teilradnabe zusammen durchschnittlich 50 Minuten.

Das Ergebnis hat gezeigt, dass eine voxelbasierte fünfachsiges NC-Simulation eines Verdichterlaufrades mit einem eingesetzten SGI-System (mit vier 150 MHz MIPS R4400 CPU unter dem Betriebssystem IRIX) nicht in Echtzeit durchgeführt werden konnte. Die vorgestellten Resultate bezogen sich auf nicht parallelisierte Testläufe.

Es ist jedoch sicher, dass die modernste Hardwaretechnik (Geschwindigkeit: ein paar GHz) zusammen mit paralleler Softwaretechnik eine echtzeitgemäße voxelbasierte fünfachsiges NC-Simulation ermöglichen kann. Die Umsetzung eines voxelbasierten fünfachsiges NC-Simulationssystems auf wesentlich preiswertere PCs unter Windows NT oder auch Linux System ist in einer zukünftigen Arbeit vorgesehen.

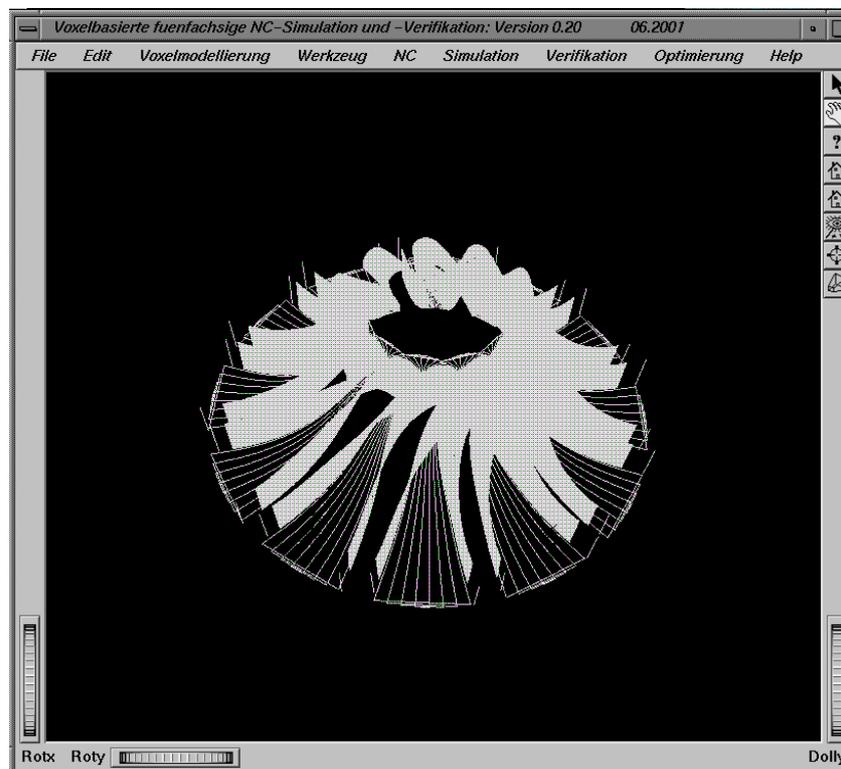
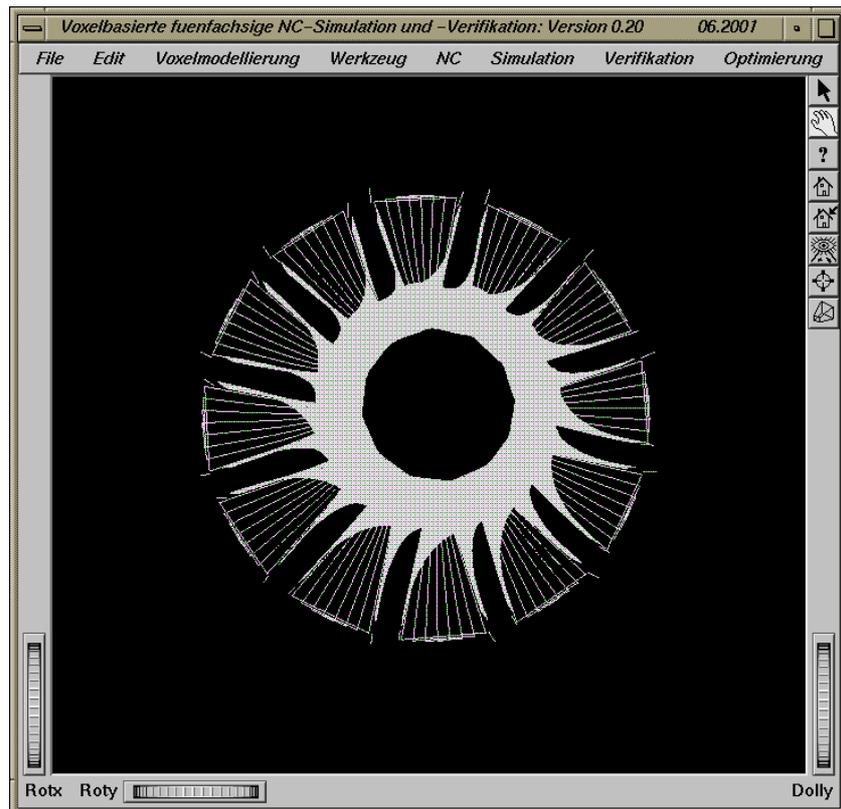


Bild 7.9: Darstellung der Fräserbahnen zur NC-Bearbeitung des vollständigen Verdichterlaufrades

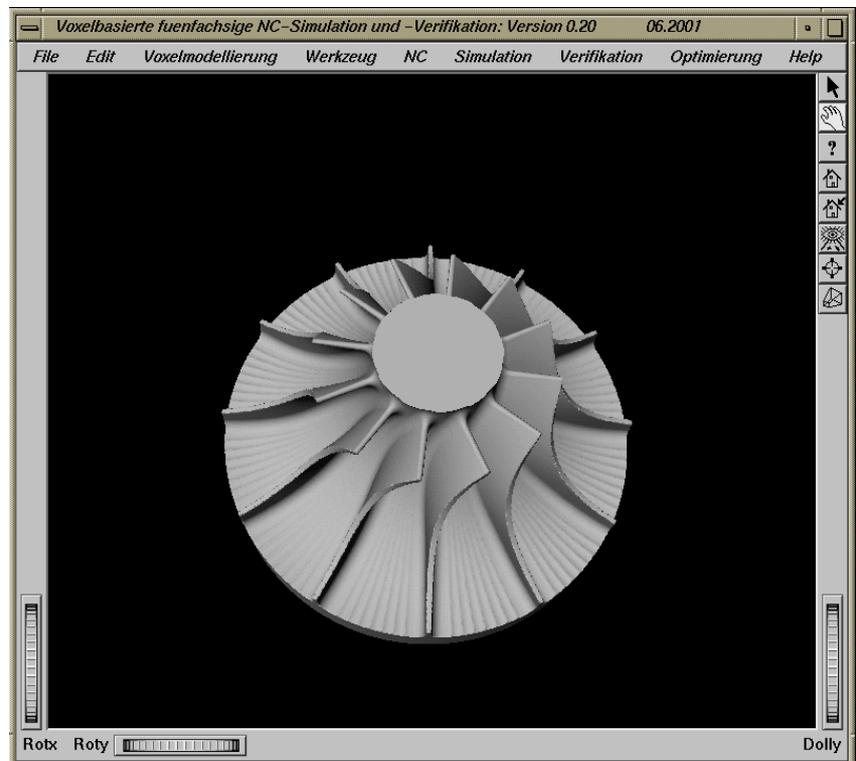
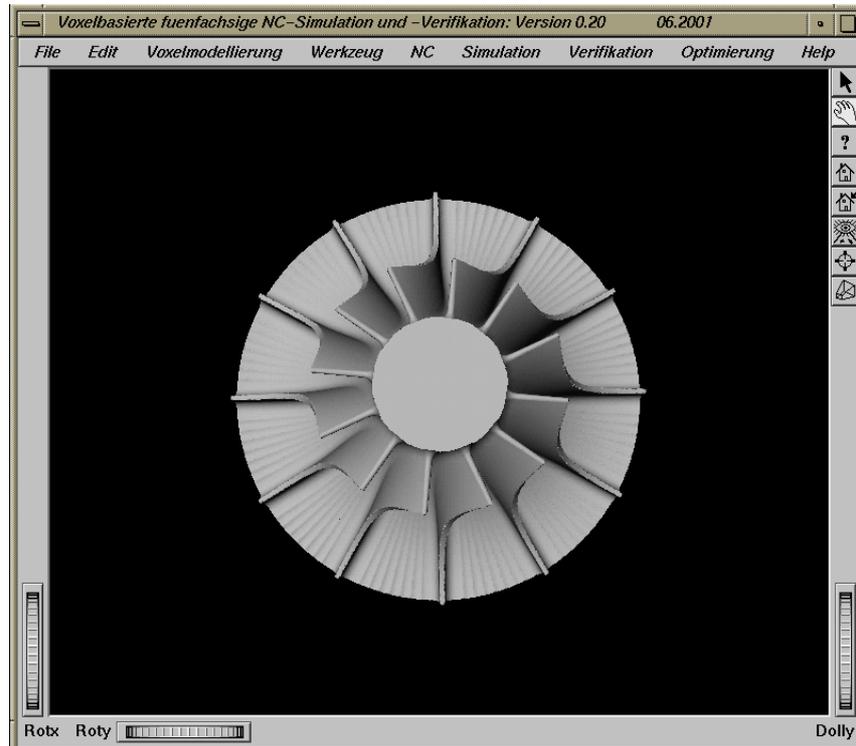


Bild 7.10: Darstellung des simulierten Verdichterlaufrades in verschiedenen Ansichten

8 Zusammenfassung

Die Simulationstechnik wird als Hilfsmittel zur Beherrschung der drei- und fünfsichtigen NC-Fräsbearbeitung eingesetzt. Sie ermöglicht detaillierte Aussagen über den NC-Fräsvorgang sowie die Optimierung eines NC-Programms vor der realen NC-Fräsbearbeitung, wodurch dem Anwender eine Vorstellung der komplizierten dynamischen NC-Fräsbearbeitung ermittelt wird und ein fehlerfreies NC-Programm hergestellt werden kann.

Die NC-Simulation realer Abläufe auf Rechenanlagen setzt die Aktualisierung der Werkstückgeometrie entsprechend den Bearbeitungsanweisungen voraus. Ausgehend von einer Analyse der existierenden Simulationsmethoden bei den vergangenen Forschungen und heutigen CAD-Systemen wurde als Defizit festgestellt, dass die stetige Aktualisierung simulierter Werkstücke heute noch nicht ausreichend unterstützt wird. Dieses Problem wird hauptsächlich durch die rechnerinterne Darstellung verursacht. Es lässt sich durch Einsatz von neuen innovativen Verfahren der rechnerunterstützten 3D-Modellierung und der erhöhten Rechenleistung lösen.

In der vorliegenden Arbeit wurde eine voxelbasierte Methode für die drei- und fünfsichtige Simulation des NC-FräSENS entwickelt, um eine stetige Aktualisierung des simulierten Werkstücks zu verwirklichen und die Informationen über Simulationsergebnisse für die Optimierung des NC-Programms zu gewinnen. Die Grundlage für den Aufbau des Simulationssystems ist der am IPK Berlin entwickelte 3D-Kernmodellierer des Virtual Clay Modelling Systems. Der Kernmodellierer zeichnet sich aus durch die echtzeitnahe Modellierbarkeit der zeitlich geänderten Formen, die in einem diskret geometrisch vollständigen Volumen als Voxelmodell abgebildet werden. Das simulierte Werkstück und das Zerspanungsvolumen werden durch ein Voxelmodell rechnerintern dargestellt. Die Werkstückaktualisierung erfolgt im geometrischen 3D-Raum durch eine einfache Boolesche Operation, welche nicht explizit durchgeführt werden kann. Danach kann die Oberfläche des simulierten Werkstücks mit dem Marching-Cube-Verfahren zum Zweck der graphischen Darstellung extrahiert werden. Dadurch können die Anwender das simulierte Werkstück aus beliebigen Richtungen betrachten, ohne neue Berechnungen der Visualisierungsdaten durchführen zu müssen. Die Materialverletzungen und die Restmaterialien werden ebenso einfach durch Vergleichen des simulierten Werkstücks und der Soll-Flächen des Werkstücks ermittelt. Die gewonnenen Informationen wie Materialverletzungen und Restmaterialien, falls sie vorhanden sind, werden mit verschiedenen Farben graphisch dargestellt und auch in entsprechenden Protokollen gespeichert. Diese Protokolle werden bei der Optimierung des NC-Programms angewandt.

Um den Simulationsergebnissen gerecht werden zu können, wurden die aufgrund der diskreten rechnerinternen Darstellung des simulierten Werkstücks verursachten Simulationsfehler eingehend analysiert. Dabei wurden die konservativen Methoden zur Kontrolle dieser Simulationsfehler wie Flächenfehler, Unterschnittfehler sowie Restmaterialfehler (siehe Abschnitt 3.4) unter Berücksichtigung des eingesetzten Fräasers entwickelt. Die abgeleiteten Diskretisierungsgrade wurden bei der Modellierung des Materialmodells im VCM-System und der Gestaltung des Zerspanungsvolumens eingesetzt. Entsprechend wurde das VCM-System erweitert.

Ein wesentlicher Schritt zur Aktualisierung des simulierten Werkstücks ist die Generierung des Zerspanungsvolumens im 3D-Raum. Dazu wurde eine voxelbasierte Methode zur Gestaltung von Zerspanungsvolumen der diversen Werkzeuge wie zylindrischen Fräser, Torus-, zylindrischen Gesenk-, Kugelkopf-, Fass-, Kegel- und kegeligen Gesenkfräser entwickelt. Dabei wurden zuerst die Grundkörper in verschiedene Fräsertypen klassifiziert und danach die Grundkörper und deren Bewegungsvolumen, basierend auf Voxelmodell, modelliert. Die Zerspanungsvolumen setzen sich aus den voxelbasierten Volumen zusammen. Der Vorteil bei dieser Methode liegt darin, dass aufgrund der diskreten Datenstruktur des Voxelmodells und der damit ermöglichten Abbildung von beliebigen Formen keinerlei Einschränkungen zur Modellierung der Zerspanungsvolumen der diversen Fräser-typen gegeben sind.

Zur Beschleunigung des Aktualisierungsprozesses wurden ausgehend von den Eigenschaften des Voxelmodells die Maßnahmen getroffen. Um das Parallelitätspotential in der Werkstückaktualisierung ausnutzen zu können, wurden die Aktualisierungsaktivitäten in solche Teilkomponenten zerlegt, die sich bei der Werkstückaktualisierung parallel durchführen lassen. Bei der Diskretisierung des Zerspanungsvolumens wurde der Arbeitsbereich in einem sogenannten 3D-Voxelraum, der das Materialmodell umhüllt, beschränkt, da das Zerspanungsvolumen des Fräasers nur in diesem Voxelraum an der Materialabtragung beteiligt ist. Die Visualisierung des simulierten Werkstücks wurde selektiv im segmentierten Subvolumen des Voxelmodells durchgeführt, in welchem die Form des Voxelmodells verändert wurde. Es wurde auch den Anwendern die Möglichkeit gegeben, die graphische Darstellung des simulierten Werkstücks zu steuern oder sogar auszublenden, um die Aktualisierungsprozesse drastisch beschleunigen zu können.

Die Überprüfung der Materialverletzungen und der verbliebenen Restmaterialien konnte direkt durch die Berechnung der Abstände zwischen dem Stixelpunkt des simulierten Werkstücks und der Sollfläche durchgeführt werden. Dazu wurde die Newton/Raphson-Methode eingesetzt. Die Soll-Flächen des Werkstücks wurden zuerst im 3D-Raum durch Stixelmodell unter Berücksichtigung der Flächenfehler (siehe Abschnitt 3.4) diskretisiert. Die Stixelpunkte der Soll-Flächen wurden je nach

deren Positionen in die Voxel-Struktur hinzugefügt, die jeweils diese Stixelpunkte umhüllt. Diese Voxel-Struktur wurde sowohl zur Lokalisierung der Soll-Flächen als auch zur Auswahl des anfänglichen Punkts für die Newton/Raphson-Methode verwendet. Dadurch konnte der NC-Verifikationsvorgang erheblich beschleunigt werden.

Mit einem aussagekräftigen Beispiel, nämlich der Simulation der NC-Fräsbearbeitung eines Verdichterlaufrades, konnte nachgewiesen werden, dass das entwickelte Simulationssystem den aufgestellten Anforderungen in allen Punkten gerecht wird. Die im Rahmen dieser Arbeit entwickelten Methoden sind wichtige Beiträge zur drei- und fünfsichtigen Simulation des NC-Fräsens. Damit können NC-Programme für die drei- und fünfsichtige NC-Fräsbearbeitung verifiziert, optimiert und eine hohe Prozessqualität vorab sichergestellt werden.

9 Anhang A Ermittlung des maximalen Unterschnittfehlers bei der voxelbasierten Repräsentation

Im Stixel-Datenformat für die Repräsentation von dreidimensionalen Objekten werden die Positionen des Materialwechsels gespeichert. Dazu werden von drei Ebenen $E_1(x, y)$, $E_2(y, z)$ und $E_3(z, x)$ senkrechte Strahlen in die Szene gesetzt. Während entlang der Strahlen die Oberflächenpositionen mit Fließkommagenauigkeit gespeichert werden können, haben diese Strahlen zwangsläufig einen diskreten Abstand zueinander. Bild 9.1 stellt einen von solchen Strahlen gebildeten Voxel-Würfel dar. H ist die Kantenlänge des Voxel-Würfels.

Die voxelbasierte NC-Simulation erfolgt durch Schneiden von Stixeln während der Bewegung des Werkzeugs. Die Oberfläche des simulierten Werkstücks wird durch die vom Marching-Cubes-Verfahren extrahierten Dreiecksnetze angenähert. Es ist möglich, dass die Dreiecksflächen ohne die Verletzungen der Oberfläche des Werkstücks in den Sample-Punkten unterschritten werden können. Der maximale Unterschnittfehler bei einem Dreieck tritt auf, wenn das Werkzeug mit den drei Eckpunkten dieses Dreieckes überlappt ist. Daraufhin wird der maximale Unterschnittfehler in einem Würfel untersucht. Gewöhnlich wird hier ein Kugelfräser ausgewählt.

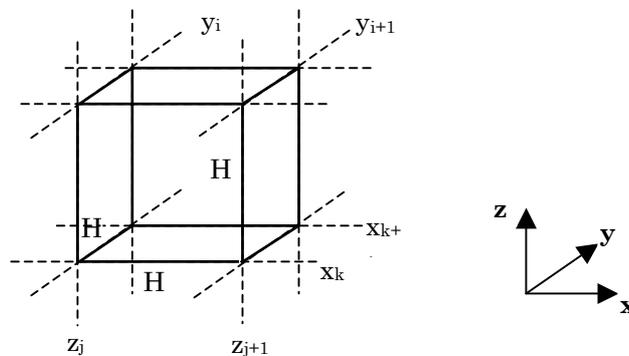


Bild 9.1: Ein Würfel im Voxelmodell

Da der maximale Unterschnittfehler in einem Würfel untersucht werden soll, werden die folgenden drei Fallunterscheidungen in einem Würfel in Betracht gezogen:

- Ein Werkzeug überlappt mit zwei Eckpunkten,
- Ein Werkzeug überlappt mit drei Eckpunkten, aber nicht durch zwei Endpunkte einer Kante des Würfels,

- Ein Werkzeug überlappt mit drei Eckpunkten, zwei davon gehören zu einer Kante des Würfels.

9.1 Überlappung des Werkzeugs mit zwei Eckpunkten

Bild 3.6 stellt die Beziehung zwischen dem Unterschnittfehler ε_u und dem Abstand zwischen zwei Punkten A und B dar. Der Unterschnittfehler ε_u wird in der folgenden Formel bestimmt:

$$\varepsilon_u = r - \sqrt{r^2 - \frac{1}{4}D^2} \quad (9-1)$$

Der maximale Abstand (D) zwischen den Eckpunkten eines Würfels im Voxelmodell liegt in der Diagonalen des Würfels. Somit ergibt sich

$$D_{\max} = \sqrt{3}H \quad (9-2)$$

In diesem Fall wird der maximale Unterschnittfehler ε_u bezeichnet durch folgende Formel

$$(\varepsilon_u)_{\max 1} = r - \sqrt{r^2 - \frac{3}{4}H^2} \quad (9-3)$$

9.2 Überlappung des Werkzeugs mit drei Eckpunkten, aber nicht durch zwei Endpunkte einer Kante des Würfels

Im Bild 9.2 wird der Fall gezeigt: Ein Werkzeug überlappt mit drei Eckpunkten eines Würfels, aber nicht durch zwei Endpunkte einer Kante des Würfels.

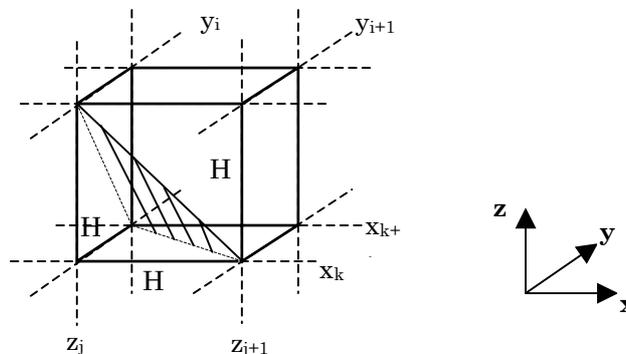


Bild 9.2: Drei Eckpunkte, die zu unterschiedlichen Kanten eines Würfels im Voxelmodell gehören

Die Beziehung zwischen D und dem schattierten gleichseitigen Dreieck wird im Bild 9.3 gezeigt.

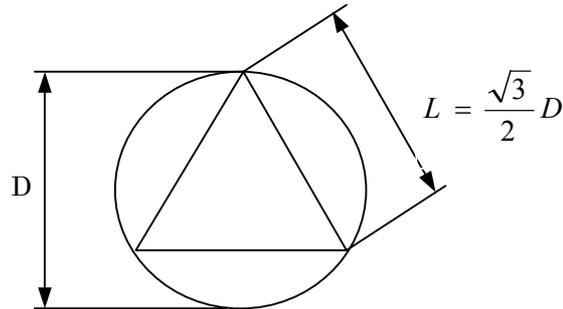


Bild 9.3: Ein Werkzeug durch ein gleichseitiges Dreieck

Nach dem Bild 9.2 ergibt sich

$$L = \sqrt{2}H \quad (9-4)$$

Dadurch wird der maximale Unterschnittfehler ε_u in diesem Fall abgeleitet.

$$(\varepsilon_u)_{\max 2} = r - \sqrt{r^2 - \frac{2}{3}H^2} \quad (9-5)$$

9.3 Überlappung des Werkzeugs mit drei Eckpunkten, aber zwei davon gehören zu einer Kante des Würfels

In diesem Fall stellen diese drei Eckpunkte ein rechtwinkliges Dreieck dar. Bild 9.4 zeigt dieses rechtwinklige Dreieck (ABC). Die zwei Katheten des Dreieckes werden durch H und βH bezeichnet, wobei $\beta = 1$ oder $\sqrt{2}$ sein kann. Falls $\beta = 1$ ist, ist diese Kathete auch Kante des Würfels im Voxelmodell. Wird $\beta = \sqrt{2}$, ist diese Kathete eine Diagonale einer Seitenfläche des Würfels. In Bild 9.4 stellt Punkt O(x, y, z) das Zentrum der Werkzeugstirnfläche dar. Es ist ersichtlich, dass

$$\begin{aligned} X^2 + Y^2 + Z^2 &= r^2 \\ (X - \beta H)^2 + Y^2 + Z^2 &= r^2 \\ X^2 + (Y - H)^2 + Z^2 &= r^2 \end{aligned} \quad (9-6)$$

wobei $X = \frac{1}{2}\beta H$, $Y = \frac{1}{2}H$, $Z = \sqrt{r^2 - \frac{1}{4}(1 + \beta^2)H^2}$.

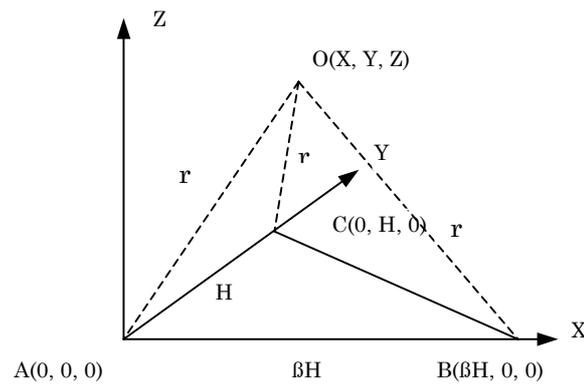


Bild 9.4: Ein Werkzeug überlappt mit drei Eckpunkten eines rechtwinkligen Dreieckes

Folglich wird der maximale Unterschnittfehler ε_u abgeleitet nach

$$(\varepsilon_u)_{ma} = r - Z = r - \sqrt{r^2 - \frac{1}{4}(1 + \beta^2)H^2} \quad (9-7)$$

wenn $\beta = 1$, ergibt sich

$$(\varepsilon_u)_{\max 3} = r - \sqrt{r^2 - \frac{1}{2}H^2} \quad (9-8)$$

wenn $\beta = \sqrt{2}$, ergibt sich

$$(\varepsilon_u)_{\max 4} = r - \sqrt{r^2 - \frac{3}{4}H^2} \quad (9-9)$$

Zusammenfassend ergibt sich von den Formeln (9-3), (9-5), (9-8), (9-9) der maximale Unterschnittfehler in einem Voxel-Würfel:

$$(\varepsilon_u)_{\max} = r - \sqrt{r^2 - \frac{3}{4}H^2} \quad (9-10)$$

10 Anhang B Beispiele von Volumenspurmodellen

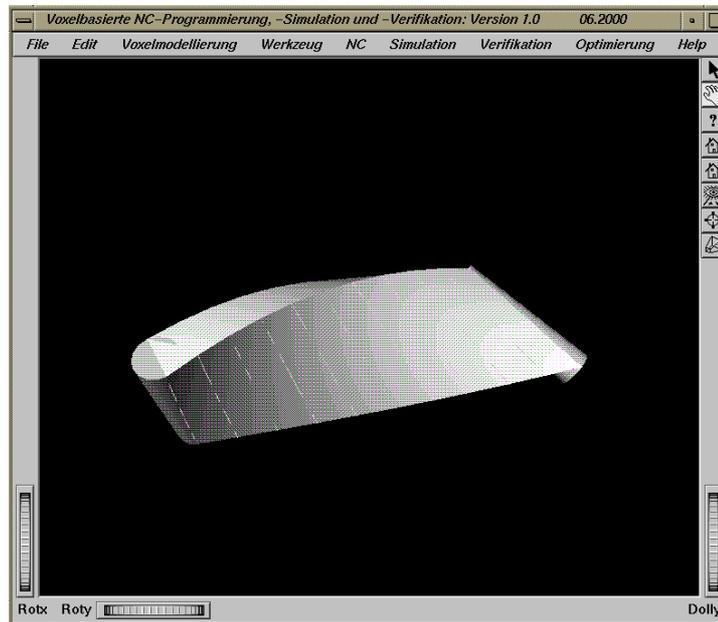


Bild B1: Volumenspurmodell eines zylindrischen Fräsers mit zwei Fräserbahnen

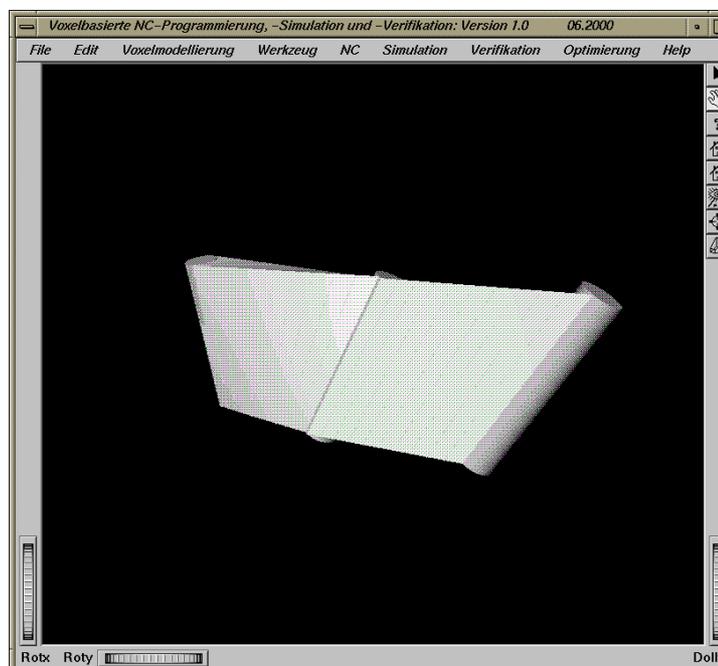


Bild B2: Volumenspurmodell eines zylindrischen Fräsers mit drei Fräserbahnen

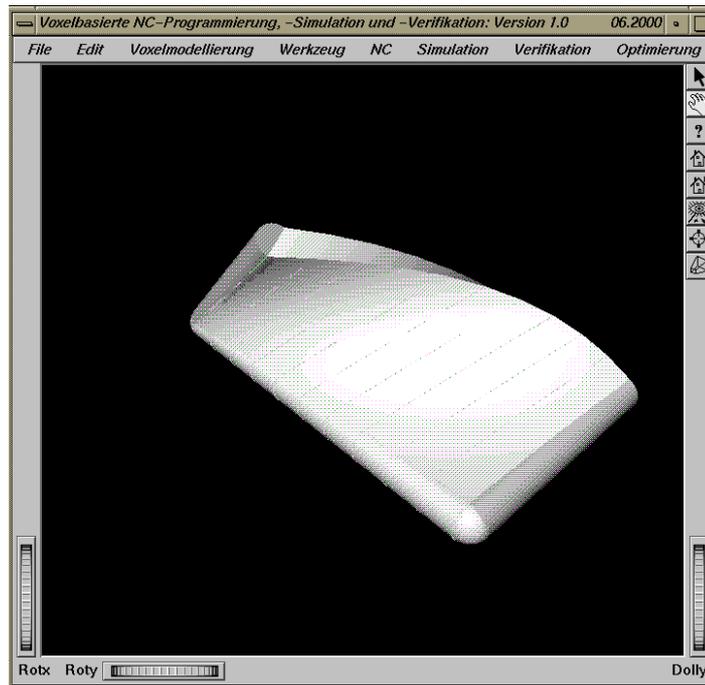


Bild B3: Volumenspurmodell eines zylindrischen Gesenkfräsers mit zwei Fräserbahnen

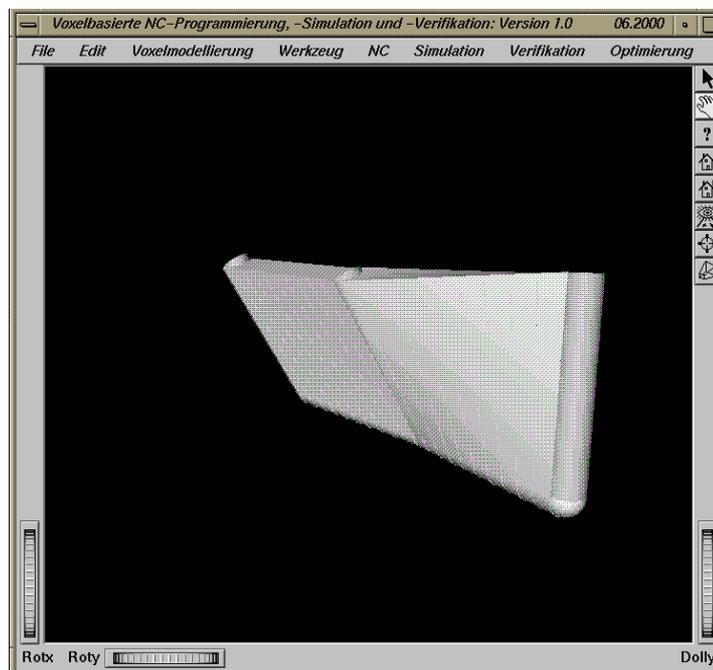


Bild B4: Volumenspurmodell eines zylindrischen Gesenkfräsers mit drei Fräserbahnen

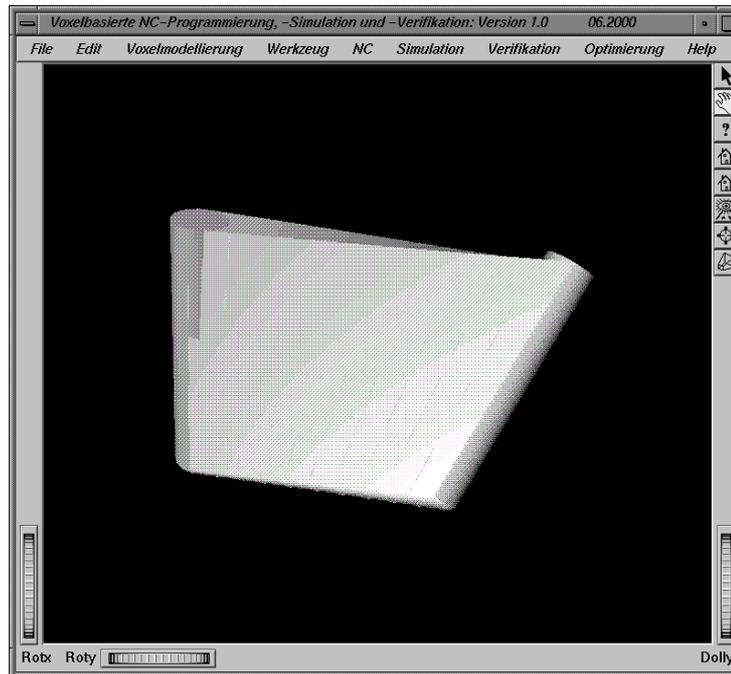


Bild B5: Volumenspurmodell eines Torusfräsers mit zwei Fräserbahnen

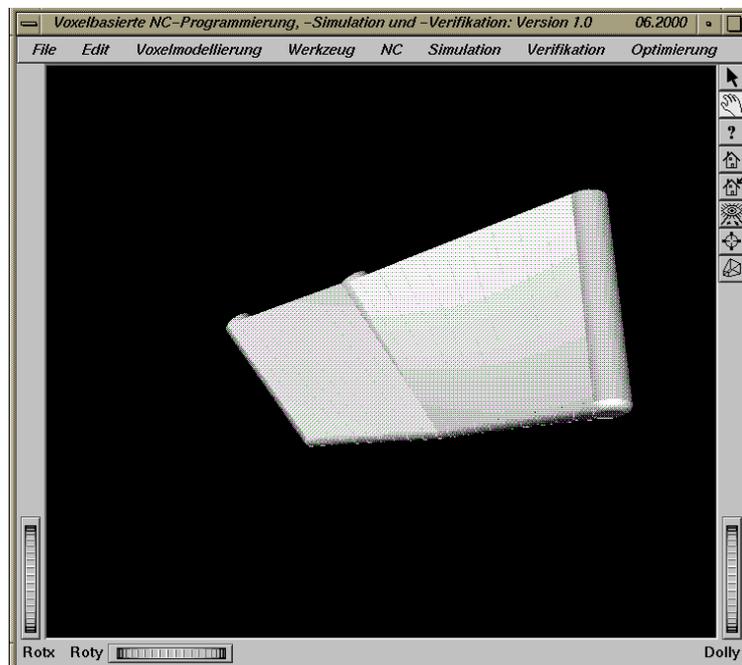


Bild B6: Volumenspurmodell eines Torusfräsers mit drei Fräserbahnen

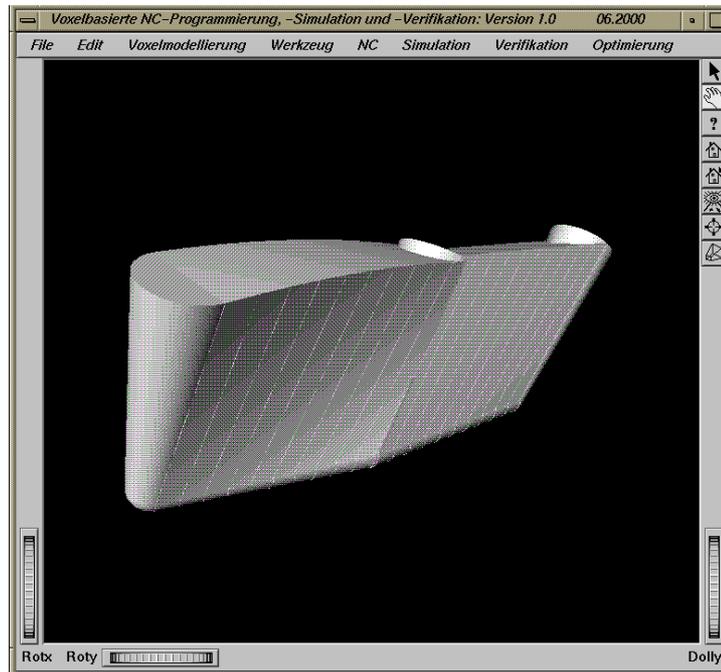


Bild B7: Volumenspurmodell eines kegeligen Gesenckfräasers mit drei Fräserbahnen

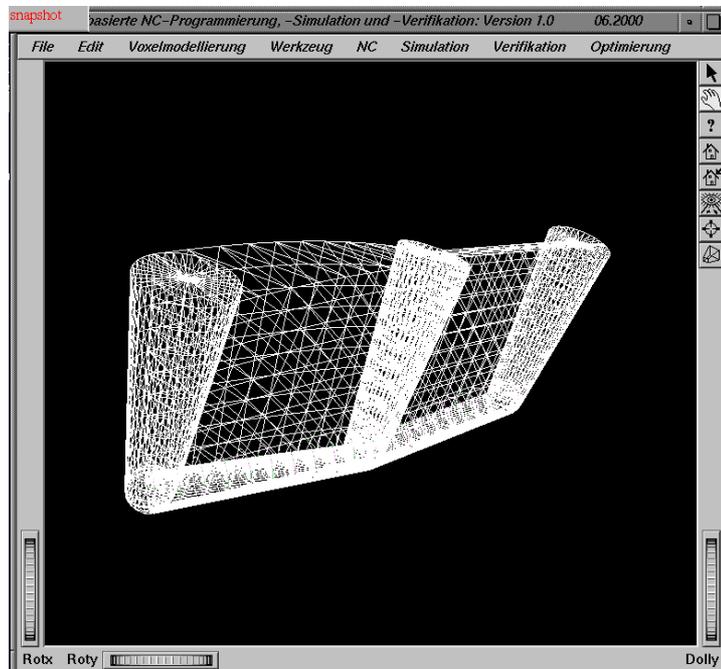


Bild B8: Volumenspurmodell von Bild B7 mit Dreiecksflächen dargestellt

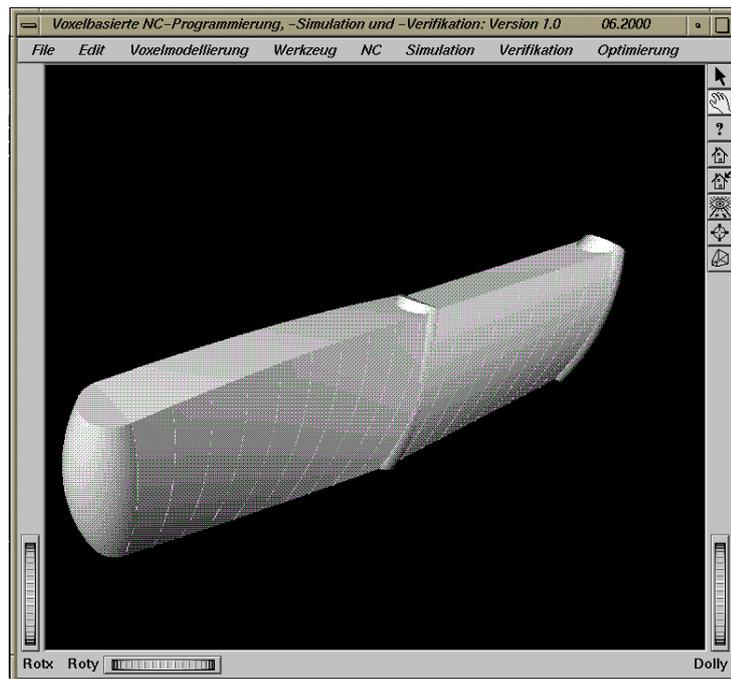


Bild B9: Volumenspurmodell eines Fassfräasers mit drei Fräserbahnen

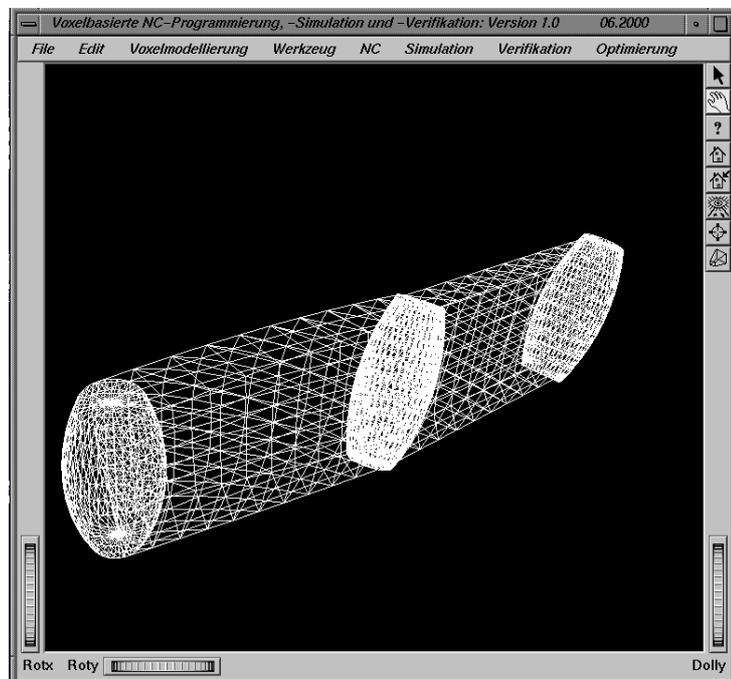


Bild B10: Volumenspurmodell von Bild B9 mit Dreiecksflächen dargestellt

11 Anhang C Beispiele von Simulationsmodell

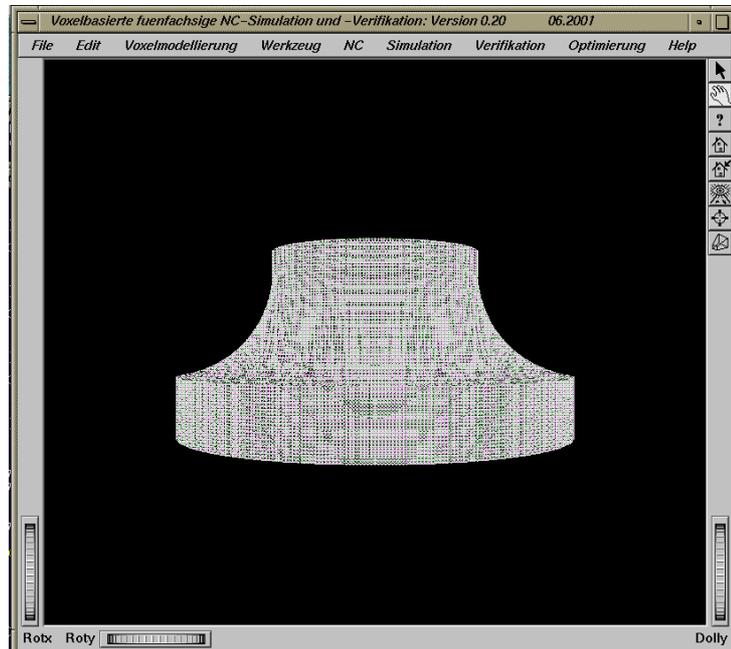


Bild C1: Voxelmodell des Materialmodells

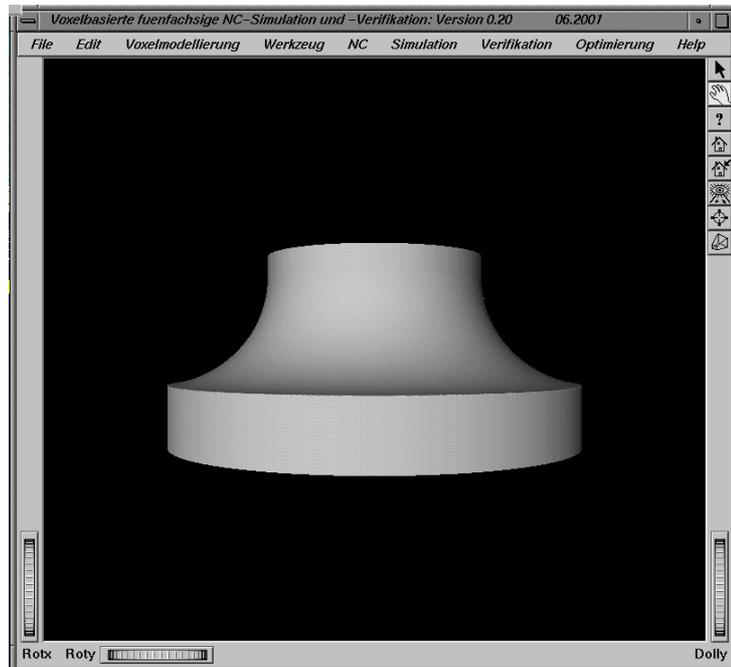


Bild C2: Materialmodell

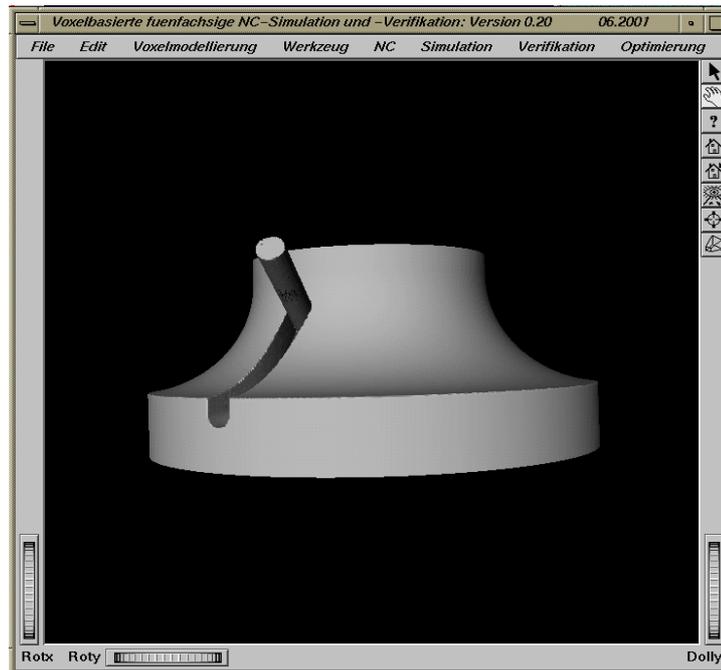


Bild C3: NC-Simulation: Materialmodell mit zylindrischem Gesenkfräser

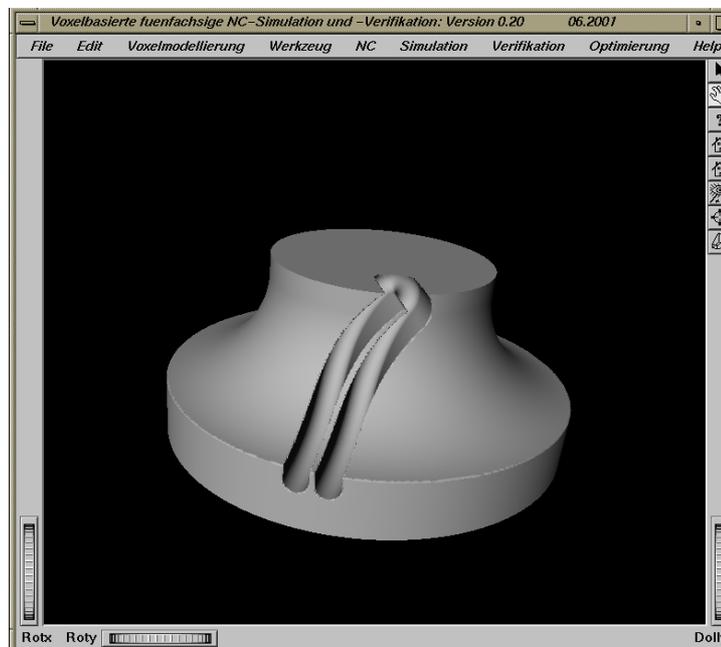


Bild C4: Werkstückmodell nach NC-Simulation einer Schaufel

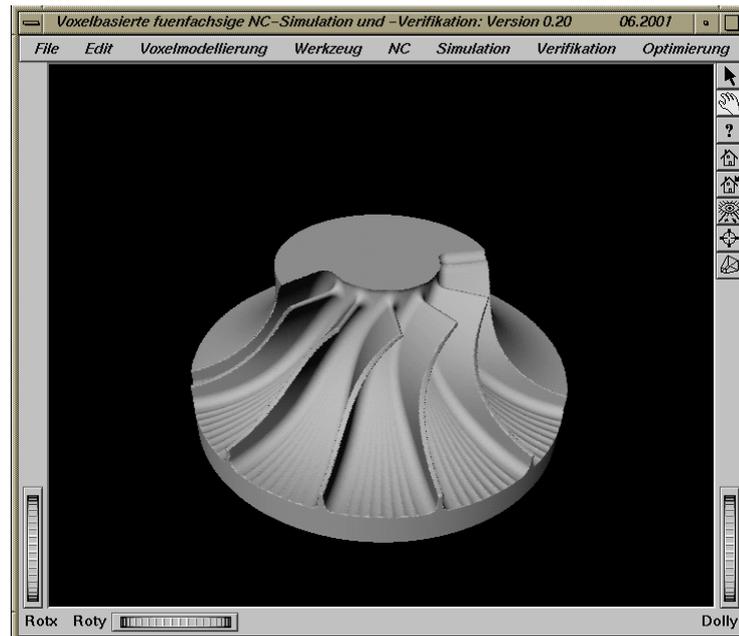


Bild C5: Werkstückmodell nach mehreren Simulationsstufen

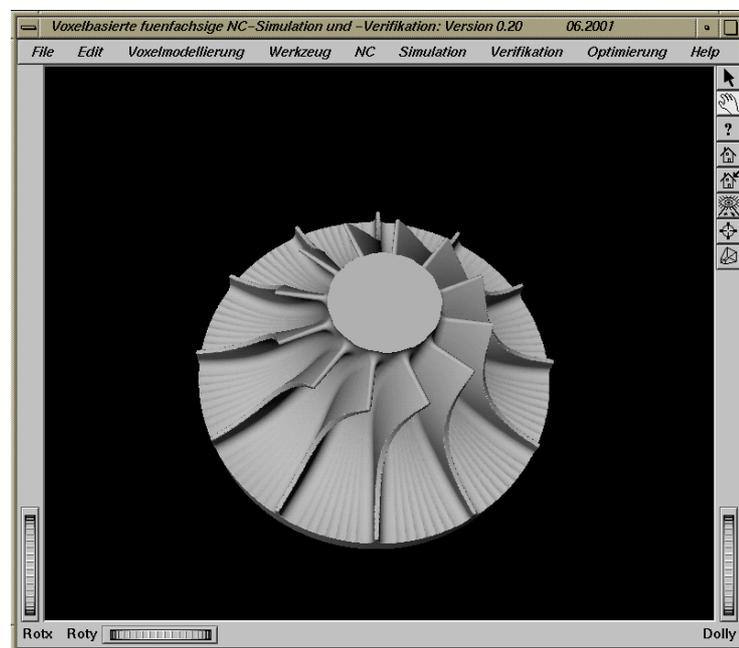


Bild C6: Vollständiges Verdichterlaufrad nach der NC-Simulation
in Schrägansicht

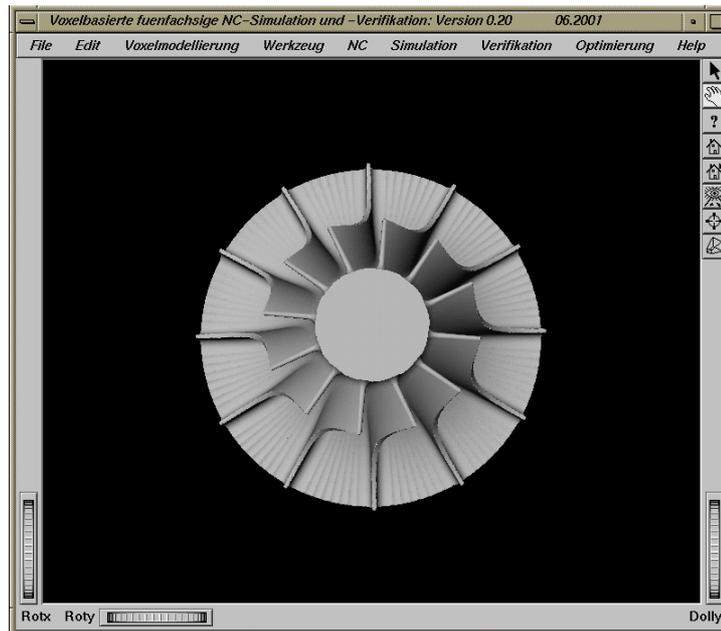


Bild C7: Draufsicht des vollständigen Verdichterlaufrades

12 Literatur

- 1 Krause, F.-L.; Lüddemann, J.: Anforderungen an die integrierte Konstruktion und Fertigung von Freiformflächen. In: wbk Karlsruher Kolloquium: Konstruktion und Fertigung von Freiformflächen, Tagungsband, Februar 1991, S.119-38
- 2 Hirsch, B.; Sheng, X.; Müller, H.: Realistische Simulation mehrachsiger NC-Bearbeitungsvorgänge. ZWF 85 (1990) 10, S.541-545
- 3 Spur, G.; Krause, F.-L.: Das virtuelle Produkt. Management der CAD-Technik. Carl Hanser Verlag, München, Wien 1997
- 4 Narvekar, A.; Huang, Y.; Oliver, J. H.: Intersection of rays with parametric envelope surfaces representing five-axis NC milling tool swept volume. In: Proceedings of ASME Advances in Design Automation, Vol.2, 1992, S.223-230
- 5 Van Hook, T.: Real Time Shaded NC-Milling Display. In: ACM SIGGRAPH '86 Computer Graphics Proceedings, Vol.20, 1986, S.15-20
- 6 Saito, T.; Takahashi, T.: NC Machining with G-Buffer Method. In: ACM SIGGRAPH '91 Computer Graphics Proceedings, Vol.25, No.4, July 1991, S. 207-216
- 7 Huang, Y.; Oliver, J. H.: NC Milling Error Assessment and Tool Path Correction. In: ACM SIGGRAPH '94 Computer Graphics Proceedings, Vol.28, 1994, S.287-294
- 8 Hui, K. C.: Solid Sweeping in image Space – Application in NC Simulation. In: The Visual Computer, Vol.10, 1994, S. 306-316
- 9 Weinert, K.; Müller, H.; Kreis, W.; Surmann, T.; Ayasse, J.; Schüppstuhl, T.; Kneupner, K.: Diskrete Werkstückmodellierung. ZWF 97 (2002) 7-8, S. 385-389
- 10 Kawashima, Y.; Itoh, K.; Ispida, T.; Nonaka, S.; Ejiri, K.: A Flexible Quantitative Method for NC Machining Verification Using a Space-Division Based Solid Model. In: The Visual Computer, Vol.7, 1991, S.149-157
- 11 Brunet, P.; Navazo, I.: Solid Representation and Operation Using Extended Octree. In: ACM Transactions on Graphics, 9, 2, April 1990, S.170-197

- 12 Menon, J. P.; Robinson, D. M.: Advanced NC Verification Via Massively Parallel Raycasting: Extensions to new Phenomena and Geometric Domains. In: ASME Manufacturing Review, 6, 2, June 1993, S.141-154
- 13 Levoy, M.: Volume Rendering: A Hybrid Ray Tracer for Rendering Polygon und Volume Data. In: IEEE Computer graphics and Applications, Vol.10, No.2, March 1990, S.33-40
- 14 Levoy, M.: Volumen Visualisation: A Look Back, A Look Ahead. Keynote address given at San Diego Workshop on Volume Visualisation, December 1990
- 15 Menon, J.; Marisa, R.J.; Zagajac, J.: More Powerful Solid Modeling through Ray Representation. In: IEEE Computer Graphics & Applications, May 1994, S.22-35
- 16 Güldenpfennig, P.M.; Wehmeyer, K.: Neue Steuerung im industriellen Einsatz. ZWF 91 (1996) 6, S.250-253
- 17 Huang, Y.; Oliver, J. H.: Integrated Simulation, Error Assessment, and Tool Path Correction for Five Axis NC Milling. In: Journal of Manufacturing Systems, Vol.14, No.5, 1995, S.331-344
- 18 Buchbauer, T.: „CAM“. In: Megatech, Vol.2, 1993, S.18-23
- 19 <http://www.cgtech.com>, Juli 2003
- 20 <http://www.dcade.de>, April 1999
- 21 <http://www.hypercadcam.com>, Juli 2003
- 22 <http://www.sescoi.de>, Juli 2003
- 23 Sungurtekin, U. A.; Voelcker, H.: Graphical Simulation & Automatic Verification of NC Milling Programs. In: Proceeding of IEEE International Conference on Robotics and Automation. San Francisco, CA., 1986, S.156-165
- 24 Lüddemann, J.: Virtuelle Tonmodellierung zur skizzierenden Formgestaltung im Industriedesign. Berichte aus dem Produktionstechnischen Zentrum Berlin, FhG/IPK-Berlin, Dissertation TU Berlin, 1996

- 25 Meier, A.: Methoden der graphischen und geometrischen Datenverarbeitung. B.G. Teubner Verlag, Stuttgart 1986
- 26 Grätz, J. F.: Handbuch der 3D CAD Technik, Modellierung mit 3D Volumensystemen. Siemens AG, Berlin, München 1989
- 27 Kopp, H.: Graphische Datenverarbeitung, Methoden, Algorithmen und ihre Implementierung. Carl Hanser Verlag, München, Wien 1989
- 28 Schmidt, W.: Graphikunterstütztes Simulationssystem für komplexe Bearbeitungsvorgänge in numerischen Steuerungen. Springer-Verlag, Berlin 1988
- 29 Kopp, H.: Graphische Datenverarbeitung, Methoden, Algorithmen und ihre Implementierung. Carl Hanser Verlag, München, Wien 1989
- 30 N.N.: TIPS-1 Technical Information Processing System. Institute of Precision Engineering, Hokkaido University, Sapporo 1978
- 31 Pritschow, G.; Ioannides, M.; Steffen, M.: Modellierverfahren für die 3D-Simulation von NC-Bearbeitungen. Teil 3. VDI-Z 135 (1993), Nr. 6, S.47-52
- 32 Tönnies, K.D.; Lemke, H.U.: 3D-Computergrafische Darstellungen. Oldenbourg Verlag, München, Wien 1994
- 33 Jense, G. J.: Voxel-Based Methodes for CAD. In: Computer Aided Design, Vol.21, No.8, 1989, S.528-533
- 34 Stifter, S.: Simulation of NC Machining Based on the Dixel Model: A Critical Analysis. In: The International Journal of Advanced Manufacturing Technology. Vol.10, 1995, S.149-157
- 35 Stifter, S.: Simulation and verification of NC machining using symbolic computation techniques. In: Proceedings IMACS 91 Congress on Computation and Applied Mathematics, Dublin, Ireland, July 1991, S. 22-26
- 36 Mayr, H.; Stifter, S.: Off-line generation of error-free robot/NC code using simulation and automated programming techniques. In: Proceedings of International Conference on CAD/CAM and AMT in Israel, North Holland, December 1989

- 37 Montani, C.; Scopigno, R.: Rendering Volumetric Data Using the STICKS Representation Scheme. In: Computer Graphics, Vol.24, No.5, November 1990, S.87-93
- 38 Frey, D.D.; Otto, K.N.; Pflager, W.: Swept Envelopes of Cutting Tools in Integrated Machine and Workpiece Error Budgeting. In: Annals of the CIRP, Vol.46, Berlin 1997
- 39 Fridshal, R.; Cheng, K.P.; Duncan, D.; Zucker, Z.: Geometric Modelling Based Numerical Control Part Program Verification. In: Proceedings of the Numerical Control Society: 19th Annual Meeting and Technical Conference, 1982
- 40 Wang, W.; Wang, K.K.: Geometric modeling for swept volumes of moving solids. In: IEEE Computer Graphics & Application. Vol.12, 1986, S.8-17
- 41 Narvekar, A.P.: Representation and application of swept solid for numerically controlled milling. M.S. Thesis. Dep. Of Mech. And Aero. Engr., State Univ. of New York at Buffalo, USA, 1990
- 42 Sambandan, K.: Geometry generated by sweeps of polygons and polyhedra. Tech. Rep. 66, Cornell Injection Modeling Project, College of Engineering, Cornell University, USA, 1990
- 43 Ganter, M.; Uicker, J.: Dynamic collision detection using swept solids. In: ASME J. Mechanism, Transmission & Automation in Des..., Vol.108, 1986, S.549-555
- 44 Leu, M.C.; Park, S.H.; Wang, K.K.: Geometric representation of translation swept volumes and its application. In: ASME J. of Eng. Indust, Vol.108, 1986, S.145-149
- 45 Weld, J.D.; Leu, M.C.: Geometric representation of swept volumes with applications to polyhedral object. In: Int. J. of Robot. Res, Vol.9, 1991, S.675-687
- 46 Kieffer, J.; Litvin, T.: Swept volume determination and interference detection for moving 3-D solids. In: ASME J. Mech. Des., Vol.113, 1991, S.456-453
- 47 Ling, Z.K.; Chase, T.: Generating the area of a body undergoing planar motion. In: J. Mech. Des.(to appear) 1995

- 48 Blackmore, D.; Leu, M.C.: A differential equations approach to swept volume. In: Proc. Rensselaer's 2nd Int. Conf. On Comput. Integrated Manufacturing, Washington, DC: IEEE Comput. Soc. Press, 1990, S.143-149
- 49 Blackmore, D.; Leu, M.C.: Analysis of swept volume via Lie groups and differential equations. In: Int. J. Robotics Res., Vol.11, 1992, S.516-537
- 50 Blackmore, D.; Leu, M.C.; Wang, L.P.; Jiang, H.: Swept Volume: A Retrospective and Prospective View. In: Neural, Parallel & Scientific Computations, Vol.5, 1997, S.81-102
- 51 Do Carmo, M.: Differential geometry of curves and surfaces. New York: Marcel-Dekker, 1976
- 52 Guilleman, V.; Pollack, A.: Differential topology. Englewood Cliffs: Prentice-Hall 1974
- 53 Warner, F.: Foundation of differential manifolds and Lie groups. Glenview: Scott Foresman 1971
- 54 Blackmore, D.; Leu, M.C.; Wang L.P.: The sweep-envelope differential equation algorithm and its application to NC machining verification. In: Computer Aided Design, Vol. 29, No. 9, 1997, S.629-637
- 55 Blackmore, D.; Leu, M.C.; Shih, F.: Analysis and Modeling of deformed swept volumes. In: Computer Aided Design, Vol.26, 1994, S.315-326
- 56 Drysdale, R. L.; Jerard, R. B.; Schaudt, B.; Hauck, K.: Discrete Simulation of NC Machining. In: Algorithmica, Vol.4, No.1, 1989, S.33-60
- 57 Elvins, T.T.: A Survey of Algorithms for Volume Visualization. In: Computer Graphics, Vol.26, No.3, August 1992, S.194-201
- 58 Kaufman, A.E.; Cohen, D.; Yagel, R.: Volume Graphics. In: IEEE Computer, July 1993, S.51-64
- 59 Drebin, R. A.; Carpenter, L.; Hanrahan, P.: Volume Rendering. In: ACM SIGGRAPH '88 Computer Graphics Proceedings, Vol.22, No.4, August 1988, S.65-74

- 60 Levoy, M.: Display of Surfaces from Volume Data. In: IEEE Computer Graphics and Application, Vol.8, No.3, May 1988, S.29-37
- 61 Van Hook, T.: Real-Time Shaded Milling Display. In: ACM SIGGRAPH '86 Computer Graphics Proceedings, Vol.20, No.4, 1986, S.15-20
- 62 Herman, G. T.; Liu, H. K.: Three dimensional display of Human Organs from Computed Tomograms. In: Computer Graphics and Image Processing. Vol.9, No.1, January 1979, S.1-21
- 63 Cline, H. E.; Lorensen, W. E.; Ludke, S.; Crawford, C. R.; Teeter, B. C.: Two Algorithms for Three dimensional Reconstruction of Tomograms. In: Medical Physics, Vol.15, No.3, May/June 1988, S.320-327
- 64 Cline, H. E.; Ludke, S.; Lorensen, W. E.; Teeter, B. C.: A 3D Medical Imaging Research Workstation. In: Volume Visualization Algorithms and Architectures, ACM SIGGRAPH '90 Course Notes, Course Number 11, ACM Press, August 1990, S.243-255
- 65 Elvins, T.T.: A Survey of Algorithms for Volume Visualization. In: Computer Graphics, Vol.26, No.3, August 1992, S.194-201
- 66 Wyvill, G.; McPheeters, C.; Wyvill, B.: Data Structure for Soft Objects. In: The Visual Computer, Vol.2, No.4, August 1986, S.227-234
- 67 Lorensen, W. E.; Cline, H. E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In: Computer Graphics, Vol.21, No.4, July 1987, S.163-169
- 68 Krause, F.-L.; Germer, H.-J.; Rieger, R.; Trebo, D.: 3D-CAD-Bausteine für die Simulation von Bearbeitungsabläufen. ZWF 85 (1990) 8, S.435-438
- 69 Koepfer, T.; Schrüfer, N.: Simulation mit Cosima. VDI-Z 132 (1990) 4, S.18-25
- 70 Schrüfer, N.: Erstellung eines 3D-Simulationssystems zur Reduzierung von Rüstzeiten bei der NC-Bearbeitung. Springer-Verlag Berlin, Heidelberg, New York, London, Paris, Tokyo, Hongkong, Barcelona, Budapest 1992
- 71 Potthast, A.; Kwok, S.H.: 3D-Simulationssystem für die Bohr- und Fräsbearbeitung. ZWF 86 (1991) 6, S.286-290

- 72 Spur, G.: 3D-Bewegungssimulation – neuartige Vorgehensweise zur Werkzeugmaschinenentwicklung. VDI-Z 132 (1990) 1, S.10-13
- 73 Hsu, P.-L.; Yang, W.-T.: Realtime 3D Simulation of 3-Axis Milling Using Isometric Projection. In: Computer Aided Design, Vol.25, No.4, April 1993, S.377-386
- 74 Friedhoff, J.: Simulationsgestützte Prozessoptimierung beim Fräsen von Freiformflächen. In: Tagungsband zum Erfahrungsforum Werkzeug- und Formenbau, Darmstadt, Dortmund, Dresden, 1998
- 75 Konold, T.: Optimierung des Fünffachsfräsens durch eine Kennzahlenunterstützte CAM-Umgebung. Diss. Universität Karlsruhe (TH), 1999
- 76 Voelcker, H. B.; White, P. R.: The Role of Solid Modeling in Machining – Process Modeling and NC Verification. SAE Technical Paper 810195, 1981
- 77 Hunt, W. A.; Voelcker, H. B.: An Exploratory Study of Automatic Verification of Programs for Numerically Controlled Machine Tools, Production Automation Project Tech. Memo. No. 34, University of Rochester, January 1982
- 78 Fridshal, R.; Cheng, K. P.; Duncan, D.; Zucker, W.: Numerical Control Part Program Verification System. In: Proceedings of Conference on CAD/CAM Technology in Mechanical Engineering, MIT Press, Cambridge, MA, March 1982, S.236-254
- 79 Chappel, I. T.: The use of vectors to simulate materialremoved by numerically controlled Milling. In: Computer Aided Design, Vol.15, No.3, 1983, S.156-158
- 80 Drysdale, R. L.; Ozair, K.; Jerard, R. B.: Diskrete Surface Representations for Simulation, Verification, and Generation of Numerical Control Programs. In: Proc. 1991 NSF Design & Manufacturing Systems Conf. Austin, TX, USA, Jan 1991, S. 9-11
- 81 Li, S. X.; Jerard, R. B.: 5-Axis Machining of Sculptured Surfaces with a Flat-End Cutter. In: Computer Aided Design, Vol.26, No.3, March 1994, S.165-178

- 82 Austin, S. P.; Jerard, R., B.; Drysdale, R. L.: Comparison of Discretization Algorithms for NURBS Surfaces with Application to numerically controlled Machining. In: Computer Aided Design, Vol.29, No.1, 1997, S.71-83
- 83 Krause, F.-L.; Lüddemann, J.: Virtual Clay Modelling. In: IFIP WG5.2 Workshop, Airlie, May 1996, S.19-23
- 84 Krause, F.-L.; Bock, Y.; Dreher, S.: Design Evaluation of Virtual Clay Models using Selective Laser Sintering. In: SSM98 Sculptured Surface Machining Conference "Machining Impossible Shapes", IFIP TC5/WG5.3 Working Conference, Chrysler Technology Center, Auburn Hills, Michigan, USA, November 9-11 1998
- 85 Shu, R.; Zhou, C.; Kankanhalli, M.: Adaptive Marching Cubes. In: The Visual Computer, Vol.11, 1995, S.202-217
- 86 Beier, K.-P.; Chen, Y.: Highlight-Line Algorithm for Realtime Surface-Quality Assessment. In: Computer Aided Design, Vol.26, No.4, April 1994, S.268-275
- 87 Faux, I.D.; Pratt, M. J.: Computational Geometry for Design and Manufacture. Ellis Horwood Limited, Chichester, UK 1979
- 88 N.N.: DIN 66215 Cutter Location Data (CLDATA). Beuth Verlag, Berlin, Köln 1974
- 89 Pritschow, G.; Ioannides, M.; Steffen, M.: Modellierverfahren für die 3D-Simulation von NC-Bearbeitungen. VDI-Z 135 (1993) 4, S.36-40
- 90 Abramowsky, S.; Müller, H.: Geometrische Modellierer. B.I. Wissenschaftsverlage, Mannheim, Wien, Zürich 1991
- 91 Weck, M.: Werkzeugmaschinen. Band 1: Maschinenarten, Bauformen und Anwendungsbe-
reiche. VDI-Verlag, Düsseldorf 1988
- 92 Spur, G.; Stöferle, T.: Handbuch der Fertigungstechnik. Band 3/1 Spannen. Carl Hanser
Verlag, München, Wien 1979
- 93 Spur, G.; Stute, G.; Weck, M.: Rechnergeführte Fertigung. Carl Hanser Verlag, München,
Wien 1977

- 94 Kief, H. B.: NC-Handbuch. NC-Handbuch Verlag, Michelstadt 1987
- 95 Potthast, A.: Numerische Steuerungen – Grundlage zur Automatisierung von Werkzeugmaschinen. ZWF Sondernummer 83, 28. Oktober 1988, S.42-44
- 96 N.N.: DIN 66025, Programmaufbau für numerisch gesteuerte Arbeitsmaschinen. Beuth Verlag, Berlin, Köln 1982
- 97 Onosato, M.; Iwata, K.: Development of a Virtual Manufacturing System by Integrated Product Models and Factory Models. In: Annals of the CIRP, Vol.42/1/1993, S.475-478
- 98 Kimura, F.: Product and Process Modelling as a Kernel for Virtual Manufacturing Environment. In: Annals of the CIRP, Vol.42/1/1993, S.147-150
- 99 Trebo, D.: Aufbau eines Werkstückinformationssystems für die simulationsunterstützte Planung von Bearbeitungsaufgaben. Reihe Produktionstechnik Berlin, Band 156, Carl Hanser Verlag, München, Wien, Dissertation TU Berlin 1994
- 100 Müller, H.; Stark, M.: Adaptive generation of Surfaces in Volume Data. In: The Visual Computer, 9(4), January 1993, S.182-199
- 101 Shu, R.; Zhou, C.; Kankanhalli, M. S.: Adaptive Marching Cubes. In: The Visual Computer, 11(4), 1995, S.202-217
- 102 Scheifler, R. W.; Gettys, J.: The X Window System. In: ACM Transactions on Graphics 5 (1986) 2, S.79-109
- 103 Scheifler, R. W.; Gettys, J.; Newman, R.: X Window System - C Library and Protocol Reference. Digital Press, New York 1990
- 104 Newman, W. M.; Sproull, R. F.: Principles of Interactive Computer Graphics. McGraw Hill, New York 1979
- 105 Mäntylä, M.: An Introduction to Solid Modeling. Computer Science Press, Rockville, Maryland 1988

- 106 Barnhill, R. E.; Kersey, S. N.: A Marching Method for Parametric Surface / Surface Intersection. In: Computer Aided Geometric Design, Vol.7, 1990, S.257-280
- 107 Mortenson, M. E.: Geometric Modeling. John Wiley & Sons Inc., New York 1985
- 108 Pegna, J.; Wolter, F.-E.: Designing and Mapping Trimming Curves on Surfaces Using Orthogonal Projection. In: Proceeding of ASME advances in Design Automation, Computer Aided and Computational Design, Vol.23-1, 1990, S.235-245
- 109 Chang, K. Y.; Goodman, E. D.: A Method for NC-Tool Path Interference Detection for a Multi-Axis Milling System. In: ASME Control of Manufacturing Process, DSC-Vol. 28/PED-Vol.52, 1991, S.23-30
- 110 Foley, J. D.; Van Dam, A.; Feiner, S. K.; Hughes, J. F.: Computer Graphics Principles and Practice. Addison-Wesley Publishing Company, New York 1990
- 111 Stroustrup, B.: Die C++ Programmiersprache. Addison Wesley 1991
- 112 Corney, J.; Lim, T.: 3D Modeling with ACIS. Saxe-Coburg Publications, Dun Eglais, Station Brae, Kippen Stirling, UK 2001
- 113 Wernecke, J.: The Inventor Toolmaker, Extending Open Inventor. Release 2, Addison Wesley, Reading, Massachusetts 1994
- 114 N.N.: IRIS Inventor Programming Guide. Silicon Graphics Inc., Mountain View, California 1992
- 115 Wernecke, J.: The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor. Release 2, Addison-Wesley Publishing Company, U.S.A. 1994
- 116 N.N.: Open Inventor™ C++ Reference Manual. The official Reference Document for Open Inventor, Release 2, Addison-Wesley Publishing Company, U.S.A. 1994
- 117 Foley, J.D.; Van Dam, A.: Fundamentals of Interactive Computer Graphics. Reading – Massachusetts: Addison-Wesley Publishing Company 1984

- 118 ISO 10303-11: Industrial Automation Systems and Integration – Product Data Representation and Exchange. Part 11, Description Methods: The EXPRESS Language Reference Manual, ISO 1994, Genf 1994
- 119 Neider, J; Davis, T.; Woo, M.: OpenGL Programming Guide. Reading - Massachusetts: Addison-Wesley 1993
- 120 <http://www.vr6power.de/laderkonzepte.htm>, Juli 2003