

Vergleich von Dateiformaten für 3D-Modelle

CEDIFA Arbeitsbericht 7

13.05.2014

Heckner Heiko, Marco Wirth

Zusammenfassung

Die Basis des „3D-Drucks“ im Bereich der additiven Fertigungsverfahren ist eine vollständige Beschreibung der Dateioberfläche in einer computergenerierten CAD-Datei. Die vorliegende Arbeit gibt einen Überblick über gängige Dateiformate und vergleicht diese miteinander. Dabei wird zunächst darauf eingegangen wie CAD-Dateien generiert werden und wie die additive Fertigung funktioniert. Hierbei wird zunächst die Modellierung mit „Computer-Aided-Design“(CAD) und „3D-Scan“ beschrieben, bevor die Zerlegung der Datei in Schichten und das „schichtweise Drucken“ erläutert wird. Der Hauptteil der Arbeit beschäftigt sich mit der Vorstellung gängiger Dateiformate und deren Vergleich. Dabei werden Vor- und Nachteile der Formate erläutert und ein tabellarischer Überblick gegeben. In Abschnitt vier wird der „Transformationsprozess beim Slicen“ dargestellt. Die 3D-Objekte werden in Schichten zerlegt und in einem weiteren Dateiformat gespeichert, welches letztlich an den Drucker weitergegeben wird. Abschließend wird ein Ausblick auf die neueren Dateiformate „AMF“ und „3MF“ gegeben, wobei diese anhand ihrer potentiellen Rolle in der Zukunft des „3D-Drucks“ analysiert werden.

Inhaltsverzeichnis

1	Einleitung	3
2	Prozess bei der Erstellung von Objekten	4
2.1	Modellierung mittels Computer Aided Design	4
2.2	Slicen und Produktion	6
3	Vergleich von Dateiformaten für 3D-Modelle.....	8
3.1	Darstellung der Dateiformate	8
3.1.1	Standard Transformation Language (STL).....	8
3.1.2	Wavefront Object Format (OBJ).....	9
3.1.3	Polygon File Format (PLY).....	10
3.1.4	Drawing Interchange File Format (DXF).....	11
3.1.5	Virtual Reality Modeling Language (VRML)	12
3.1.6	Autodesk 3D Studio (3DS)	13
3.2	Vergleich der Dateiformate.....	14
4	Datentransformation beim Slicen.....	17
4.1	SLI-/CLI-Format.....	17
4.2	Weitere Schnittstellenformate (STL, NSF, RPDF)	17
4.3	Vergleich der Schnittstellenformate	18
5	Ausblick auf zukünftige Dateiformate	19
5.1	Additive Manufacturing File Format (AMF)	19
5.2	3D Manufacturing Format (3MF)	20
6	Zusammenfassung und Ausblick.....	21
	Literaturverzeichnis.....	22
	Über die Autoren.....	25
	Kontakt	26

1 Einleitung

Das Thema „3D-Druck“ oder auch „Rapid Prototyping“ ist derzeit stark in den Medien und verschiedenen Internetforen vertreten. Weil die Technologie immer erschwinglicher wird, ist sie nicht mehr nur Firmen vorbehalten, sondern immer mehr Privatpersonen nutzen 3D-Druck zur Verwirklichung kreativer Ideen (Fastermann, 2012, S. 1).

Der Prozess des „3D-Drucks“, welcher den „Additiven Fertigungsverfahren“ zugeschrieben wird, ermöglicht die Herstellung von Objekten, welche zuvor am Computer entworfen wurden. Die dazu notwendigen Dateien werden auf der Basis von Computer-Aided-Design (CAD) entworfen. Mittlerweile existieren Programme, die nicht nur von Experten verwendet werden können und eine Vielzahl an Dateiformaten beinhalten. Der Anwender hat die Möglichkeit ein sinnvolles Format entsprechend seiner Präferenzen und der Eigenschaften des zu fertigenden Produktes zu wählen. Die verschiedenen 3D-Dateien beschreiben ein Objekt anhand von Polygonen und Oberflächen, welche in Abhängigkeit des Formates noch zusätzlich, beispielsweise mit Material-, Farb- Transparenz- oder Textureigenschaften, versehen werden können. Dabei können diese nach unterschiedlichen Kriterien differenziert werden. Kriterien können z.B. die verwendete Speicherkapazität, die Kompatibilität zu anderen Anwendungen, die Geschwindigkeit des Bildaufbaus oder die Möglichkeit der Speicherung zusätzlicher Eigenschaften sein. Ziel dieser Arbeit ist es, einige der gebräuchlichsten Dateiformate vorzustellen und diese miteinander zu vergleichen.

Dabei wird zunächst auf den allgemeinen Entstehungsprozess eines Objektes eingegangen. Nach der Erläuterung des prinzipiellen Aufbaus einer CAD-Datei wird die Transformation der Datei in einzelne Schichten, das sog. „Slicing“, beschrieben und schließlich der eigentliche Produktionsprozess aufgezeigt.

Anschließend erfolgt die Beschreibung der einzelnen Dateiformate. Dabei wird jeweils der Aufbau der Datei erklärt und Vor- sowie Nachteile erläutert. Danach werden die vorgestellten Formate miteinander verglichen, um einen kompakten Überblick zu gewähren.

Im vierten Abschnitt werden schließlich die Dateiformate beschrieben, die später an den Drucker geschickt werden. Die in Kapitel drei beschriebenen Dateien beschreiben das Objekt als Oberfläche. Vor dem Produktionsprozess werden diese mit einem sogenannten „Slicer“ in eine Datei überführt, welche die einzelnen Schichten definiert.

Abschließend wird ein Ausblick auf zwei neuartige Dateiformate (AMF, 3MF) gegeben, die in Zukunft einen neuen Standard für 3D-Druck bieten könnten und aufgezeigt, was diese von den bisherigen Formaten unterscheidet.

2 Prozess bei der Erstellung von Objekten

Zur Herstellung eines Objektes mittels des 3D-Drucks müssen einige Voraussetzungen gegeben sein. Zuerst muss das Objekt modelliert werden, also in ein Dateiformat gebracht werden, welches mit einem 3D-Drucker kompatibel ist. Eine Datei kann entweder mittels „CAD“ erstellt oder durch einen „3D-Scan“ generiert werden. Diese Datei wird dann mittels „Slicing“ in weitere Dateien umgewandelt, die die einzelnen Ebenen des Objektes beschreiben. Schließlich werden diese einzelnen Ebenen bei der Produktion übereinander gedruckt und ergeben das fertige Produkt.

2.1 Modellierung mittels Computer Aided Design

„Die vollständige Beschreibung der Geometrie durch einen dreidimensionalen Datensatz ist die unabdingbare Voraussetzung für die generative Fertigung eines Bauteils (Gebhardt, 2007, S. 18). Die meisten industriellen Anwendungen konstruieren einen 3D-Datensatz mit Hilfe eines 3D-CAD-Programmes. Ein CAD-System nutzt Datenelemente und Datenstrukturen, um ein Bauteil zu beschreiben. Hierbei werden neben der Bauteilgeometrie u.a. auch die Werkstoffe, die Oberflächengüte und das Fertigungsverfahren erfasst. Die Wahl eines CAD-Systems bestimmt dessen Datenbasis, welche die Qualität des CAD-Modells und dessen Kompatibilität zu anderen Systemen über Schnittstellen festlegt. CAD-Modelle werden nach verschiedene Modelltypen unterschieden, wobei nicht alle Modelltypen für die generative Fertigung geeignet sind. Das Eckenmodell wird durch Punkte beschrieben und ist nicht für die generative Fertigung gedacht, sondern dient als Zwischenmodell bei der Umsetzung von Rasterdaten und 2D-CAD-Modellen in 3D-CAD-Modelle. Aufgrund seiner geringen Datenmenge ermöglicht das Kantenmodell eine schnelle Darstellung von 3D-Elementen bei niedriger Rechnerleistung. Die fehlenden Informationen über die Volumina und die exakte Lage der Flächen machen auch das Kantenmodell nahezu unbrauchbar für die generative Fertigung. Das Flächenmodell beschreibt das Bauteil über seine inneren und äußeren Begrenzungsflächen. Für jede Fläche wird ein Normalenvektor definiert. Dies ist ein Vektor, der senkrecht auf einer Geraden, Kurve, Ebene oder Fläche steht. Durch ihn wird festgehalten, ob eine Fläche dem Benutzer zugewandt ist oder nicht, was beim 3D-Druck wichtig ist um festzustellen wo sich Innenseite und Außenseite eines Objektes befinden (Fastermann, 2012, S. 170). Dadurch ist das Flächenmodell gut geeignet für die generative Fertigung. Bei dem Volumenmodell ist die Orientierung des Volumens der dazugehörigen Flächen exakt definiert, was es ideal für die generative Fertigung macht. Volumenmodelle werden in Grundkörpermodelle, Flächenbegrenzungsmodelle und Hybridmodelle unterteilt (Friedrich, 2012, S. 25). Ersteres definiert ein Bauteil durch Verknüpfung von Grundkörpern wie Quadern, Kugeln, Zylindern usw. zusammen mit Booleschen Operationen. Informationen über die einzelnen Flächen werden hier nicht gespeichert, jedoch werden die Begrenzungsflächen der Grundkörper berechnet. Flächenbegrenzungsmodelle empfehlen sich v.a. für die Beschreibung besonders komplexer Körper, da sie lediglich die Einzelflächen und die Lage des

Volumens, mit Hilfe eines auf jeder Fläche stehenden, nach außen zeigenden Normalvektors abspeichert. Das Hybridmodell arbeitet mit Elementen der Grundkörpermodelle und Flächenmodelle. Es vereint die Vorteile beider Elemente und erzeugt nahezu fehlerfreie Daten, daher ist es in herausragender Weise für generative Fertigungsverfahren geeignet (Gebhardt, 2007, S. 18ff.), (Friedrich, 2012, S. 24ff.).

Eine detaillierte Beschreibung der 3D-CAD-Modelle folgt in Kapitel 3, wenn die einzelnen Dateiformate dargestellt werden. Abbildung 1 macht deutlich, dass bei den verschiedenen Dimensionen der Elemente von Dateiformaten unterschiedliche Modelltypen verwendet werden müssen. Zudem ist dargestellt wie sich der optische Aufbau der einzelnen Modelltypen unterscheidet.

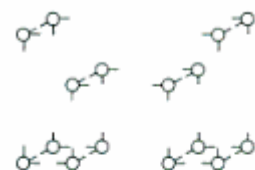
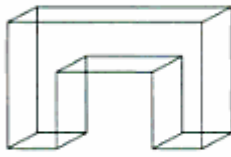
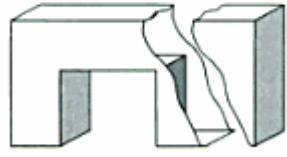
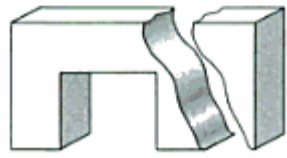
Dimension der Elemente	Element	Modelltyp
0D	<p>Punkt</p> 	Eckenmodell
1D	<p>Linie</p> 	Kantenmodell
2D	<p>Fläche</p> 	Flächenmodell
3D	<p>Volumen</p> 	Volumenmodell (Körpermodell)

Abbildung 1: CAD-Elemente und Modelltypen (Friedrich, 2012, S. 27)

Um ein Objekt via „3D-Scan“ in den Computer zu übertragen gibt es zwei Möglichkeiten. Entweder „[...] wird die Oberflächengeometrie von Objekten mit Hilfe von Laserstrahlen digital erfasst.“ (Fastermann, 2012, S. 53) oder durch einen Taster, welche maschinell oder manuell gesteuert wird. Das Ergebnis ist jeweils eine dreidimensionale Punktwolke, die eine Menge der abgetasteten Punkte darstellt. Im Gegensatz zum zweidimensionalen Scannen muss hierbei noch ein Tiefenraster zusätzlich zu dem horizontalen und vertikalen Raster ergänzt werden. Bei der kleinsten Auflösung des „3D-Scans“ wird von einem Voxel gesprochen, was

dem Pixel beim zweidimensionalen Scan entspricht. Um das Objekt in einem Grafikprogramm betrachten und bearbeiten zu können werden aus den Entfernungen der Punkte im Bezug zum Ursprung und den Winkeln Koordinaten gebildet. Alternativ wird aus der Punktwolke ein Polygonnetz gebildet, also eine geschlossene Oberfläche aus Dreiecken. Es existiert noch eine weitere Scanmethode, der „Fotoscan“, bei dem punktfokussierte Kameras in Zusammenspiel mit einem „Beamer“ die Geometrie eines Objektes anhand von Fotos erfassen (Fastermann, 2012, S. 53f.).

2.2 Slicen und Produktion

„Der 3D-Drucker arbeitet schichtweise. Dabei wird das Modell in einzelne Schichten zerlegt. Dieses Verfahren wird „Slicing“ genannt (Fastermann, 2012, S. 172). Abbildung 3 zeigt zuerst ein fertiges Volumenmodell welches dann in einem separaten Programm in Schichten geschnitten wird. Die Schichtstärke ist variabel. Durch das „Slicen“ wird eine eigene Datei erstellt, auf welche in Abschnitt vier eingegangen wird.

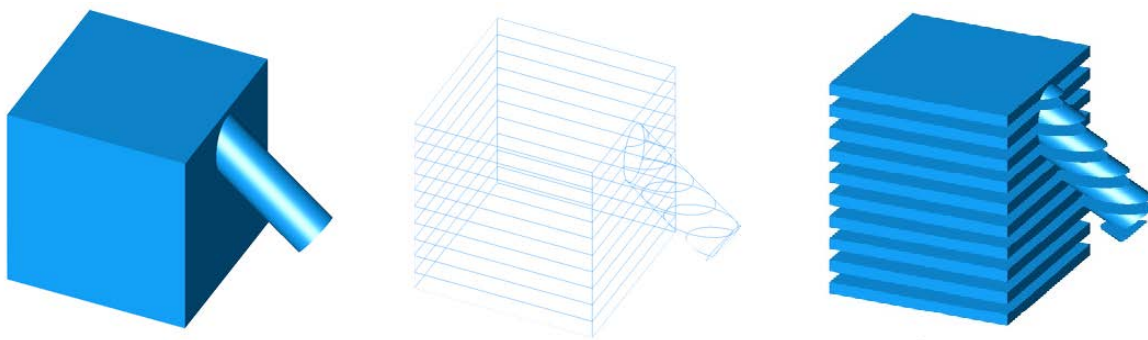


Abbildung 2: Slicen eines CAD-Objektes (Fasterpoly, 2011)

Dabei wird vorausgesetzt, dass die Ausgangsdatei fehlerfrei ist und nicht zum Beispiel Außenflächen fehlen. Die einzelnen Schichten sollen im Idealfall die gleiche Dicke haben, was jedoch nicht zwingend erforderlich ist. „Je größer die Schichtdicke ist, desto geringer ist die Oberflächenqualität. Das liegt daran, dass eine große Schichtdicke einen großen Stufeneffekt auf geneigten Oberflächen erzeugt.“ (Fastermann, 2012, S. 172) Dieser Stufeneffekt wird in Abbildung 3 anhand eines Dreiecks gezeigt. Man erkennt, dass bei einer zu groß gewählten Schichtdicke Fläche verloren geht (s. rechtes Dreieck).

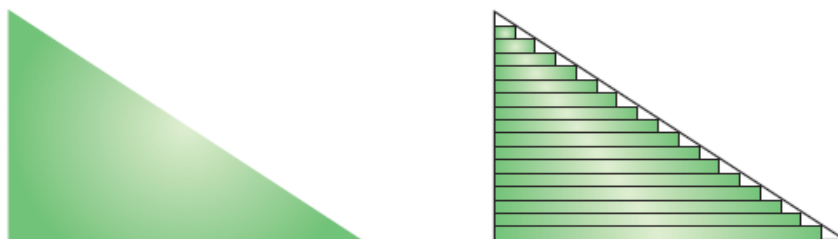


Abbildung 3: Stufeneffekt bei großer Schichtdicke (Klare & Gischer, 2009, S. 20)

Der eigentliche Produktionsprozess besteht aus einem Aufeinandersetzen dieser einzelnen Schichten. Die verwendeten Materialien hängen von dem jeweiligen Produktionsprozess ab (Gebhardt, 2007, S. 11 - 15). Es kann vorkommen, dass es in dem Modell Bereiche gibt, die zunächst keine Verbindung zu dem bereits gebauten Körper haben. In diesem Fall ist die Verwendung einer Stützstruktur notwendig, welche gesondert gefertigt werden muss. Wichtig hierbei ist, dass diese nicht mit dem Material verklebt und nach Fertigstellung leicht von dem Modell entfernt werden kann. Nach der Fertigstellung der Stützstruktur, gilt das Objekt als fertiggestellt.

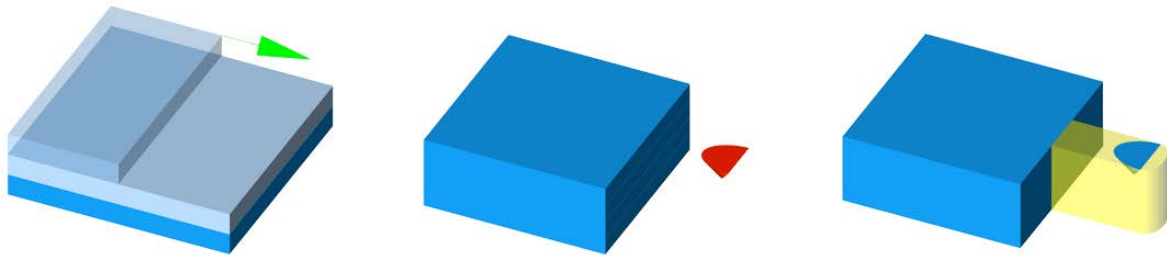


Abbildung 4: Schichtweises drucken (Fasterpoly, 2011)

In Abbildung 4 ist zunächst der Aufbauprozess der einzelnen Schichten veranschaulicht, wobei die einzelnen Schichten nur zu Demonstrationszwecken eine unterschiedliche Färbung aufweisen. In der mittleren Grafik der Abbildung 4 ist der Bereich, der ohne Stützstruktur nicht realisierbar wäre, rot gefärbt. Im rechten Teil der Abbildung ist visualisiert, wie zeitlich zum Schichtaufbau die hier gelb eingefärbte Stützstruktur hinzugefügt wird (Fasterpoly, 2011).

3 Vergleich von Dateiformaten für 3D-Modelle

Beim Vergleich verschiedener 3D-Druck-Dienstleister im Internet, fällt auf, dass einige Formate häufiger akzeptiert werden als andere. Im Folgenden werden einige der gängigsten Formate vorgestellt und anschließend miteinander verglichen.

3.1 Darstellung der Dateiformate

3.1.1 Standard Transformation Language (STL)

Die „Standard Transformation Language“, - auch „Surface Tessellation Language“ oder „Standard Triangulation Language“ genannt - ist das wohl älteste und am meisten verbreitete Dateiformat im Zusammenhang mit 3D-Druck. Es beschreibt die Darstellung der Oberfläche eines Bauteils mittels Triangulation, also durch eine Annäherung an die genaue Form durch unterschiedlich große Dreiecke (Fastermann, 2012, S. 172). Im Kontext der generativen Fertigung wird von STL als Stereolithographie-Schnittstelle gesprochen, dessen Schnittstellenformulierung schon sehr früh veröffentlicht wurde, was STL für Maschinenhersteller und unabhängige Softwarehersteller gleichermaßen nutzerfreundlich machte. „Der STL-Datensatz enthält den Normalenvektor (positive Richtung nach außen, vom Volumen weg) und die Koordinaten der drei Eckpunkte eines jeden Dreiecks“ (Gebhardt, 2007, S. 25). Er kann als ASCII-File, mit vom Menschen lesbarem Quellcode oder mit wesentlich geringerer Datenmenge als Binärfile erstellt werden. Die Dreiecksstruktur der Oberfläche ermöglicht ein Schneiden an jeder möglichen z-Koordinate, auch ohne Rückführung in das ursprünglich verwendete CAD-Programm (Danjou & Koehler, 2013). Syntaxfehler sind durch die einfache Beschreibung des Datentyps leicht zu erkennen und zu korrigieren, besonders im Vergleich zu konturorientierten Schnittstellen. Dies liegt an höherwertigen Geometrieinformationen eines Dreiecks gegenüber einem Vektor eines Konturzugs. Dennoch kann es auch hier zu Fehlern bei der Erstellung, sogenannten Konstruktionsfehlern kommen. Diese entstehen oftmals bei einer fehlerhaften Vereinigung von Einzelelementen. Dadurch entstehen oftmals überflüssige Daten. Weil bei der Beschreibung durch Dreiecke prinzipiell nur eine Annäherung an die exakte Kontur möglich ist, entstehen oft Ungenauigkeiten und Beschreibungsfehler. Abbildung 5 zeigt die Auswirkungen solcher Beschreibungsfehler. Sie können etwa zu Lücken zwischen den einzelnen Dreiecken, zu ihrer Überlappung oder gar zu einer Fehlorientierung einzelner „Dreieckspatches“ führen. Von Fehlorientierung ist die Rede, wenn der Normalenvektor in das Innere des Bauteils zeigt und Fertigungsanlagen dadurch die Innen- und Außenseite nicht mehr unterscheiden können (Gebhardt, 2007, S. 25 - 28).

Je detaillierter eine Oberfläche formuliert werden soll, desto höher ist die Anzahl an benötigten Dreiecken. Dadurch steigt die Datenmenge sehr schnell an und kann im Vergleich mit anderen Dateiformaten übermäßig viel Speicher beanspruchen (Gebhardt, 2007, S. 23 - 28).

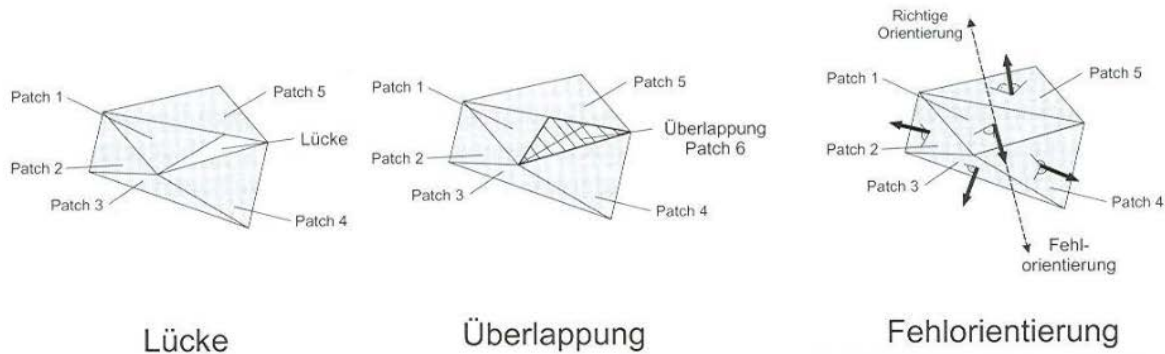


Abbildung 5: Mögliche Fehler im STL-Dateiformat (Friedrich, 2012, S. 64)

3.1.2 Wavefront Object Format (OBJ)

Das Wavefront Advanced Visualizer Format wird von fast allen Programmen akzeptiert und existiert entweder im ASCII-Format mit der Dateierweiterung „.obj“ oder im Binärformat mit der Dateierweiterung „.mod“. Im Folgenden wird das vom Menschen lesbare ASCII-Format beschrieben (Reddy, 2013). Das OBJ-Dateiformat verwendet sowohl Polygone als auch Freiform-Objekte für die grafische Darstellung. Polygonale Geometrie benutzt Punkte, Linien und Faces um eine Form zu beschreiben, Freiform-Objekte hingegen greifen auf Kurven und Oberflächen zurück.

Eine Datei beginnt üblicherweise mit einem Kommentar der durch eine voranstehende Raute gekennzeichnet ist und das Dateiformat enthält. Am Anfang objektbezogener Zeilen steht ein Schlüssel welcher die jeweilige Anweisung beschreibt. Um beispielsweise ein Polygon zu beschreiben, werden zunächst die verwendeten Eckpunkte anhand ihrer Koordinaten aufgezählt. Die Zeilen dafür beginnen mit einem „v“ für vertex und werden fortlaufend, bei eins beginnend durchnummeriert. Anschließend werden die Polygone, beginnend mit einem „f“ für faces, - aus einer Reihenfolge der Punkte gebildet. Optional können noch Texturkoordinaten (vt), Normalen (vn) und Raumeigenschaften (vp) ergänzt werden (Diemke, 2010, S. 5ff.). Ein Dreieck wird also in der Form:

```
f v/vt/vn v/vt/vn v/vt/vn
```

dargestellt. Sind keine Texturkoordinaten vorhanden, wird „vt“ durch einen zusätzlichen „Slash“ ersetzt. Fällt daneben auch der Normalenvektor weg, werden nur die Punktindizes ohne „Slash“ geschrieben. In Abbildung 6 wird der Aufbau eines einfachen Dreiecks gezeigt.

```
v 0.0 0.0 0.0
v 1.0 0.0 0.0
v 0.0 1.0 0.0

vn 0.0 0.0 1.0
vn 0.0 0.0 1.0
vn 0.0 0.0 1.0

f 1//1 2//2 3//3
```

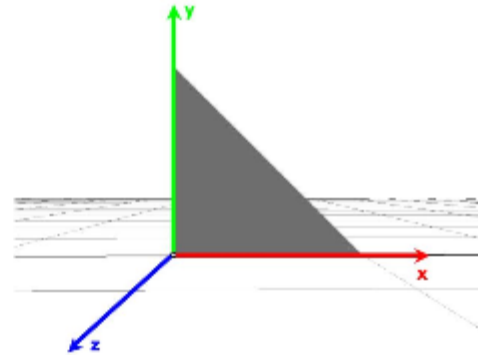


Abbildung 6: Einfaches Dreieck im .obj-Format (Schwanecke, 2010, S. 2)

Mit Hilfe des Schlüssels „g“ können Gruppen gebildet werden. Dies ermöglicht z.B. die Kombination verschiedener Polygone und ihre Speicherung in einem Unterobjekt für ihre Wiederverwendung, wobei jedes Polygon gleichzeitig zu mehreren Gruppen gehören kann. Charakteristika wie Farben, Transparenz, Lichtbrechung oder Materialien können aus einer eigenen Material-Bibliothek-Datei referenziert werden. Diese „mtl“-Datei kann mit dem Befehl „usemtl“ geladen werden, worauf die Attribute dieses Material den folgenden Flächen bis zu dem nächsten „usemtl“-Befehl zugeordnet werden (Diemke, 2010, S. 8f.).

3.1.3 Polygon File Format (PLY)

Das Polygon-File-Format, auch bekannt als Stanford Triangle Format, zeichnet sich durch eine leichte Implementierung bei flexibler Darstellungsmöglichkeit von 3D-Modellen aus. Neben der Geometrie des Objektes können hier auch weitere Daten (z.B. Kanten, Zellen, Farbwerte oder Materialien) gespeichert werden. Des Weiteren besteht die Möglichkeit innerhalb der Datei Bilddateien zu referenzieren (Danjou & Koehler, 2013).

Eine Datei im PLY-Format besteht aus einem Kopf und einem Körper und kann entweder in dem vom Menschen lesbaren ASCII oder in dem platzsparenden Binärformat gespeichert werden. Der Kopf beinhaltet neben dem Dateiformat den Namen der Datei, gefolgt von einer Auflistung der Elemente, welche noch mit zusätzlichen Eigenschaften, wie z.B. der Farbe versehen werden können. Die Auflistung der Elemente erfolgt nach einem festen Schema. So wird zuerst die Anzahl der verwendeten Eckpunkte (Vertex; Plural: Vertices) benannt, gefolgt von deren Eigenschaften (x-,y-,z-Wert und gegebenenfalls Farbwert). Daran schließt die Anzahl der Flächen („Faces“), wiederum gefolgt von deren Eigenschaftswerten. Diese bestehen aus in den Vertex-Indizes übergebenen Verweisen auf die Vertex-Liste. Es wird also redundanzfrei beschrieben, welche Eckpunkte zu welcher Fläche gehören (Danjou & Koehler, 2013). Es können im Kopf noch weitere Eigenschaften übergeben werden, wie z.B. die Flächennormale, die Transparenz, Kanten, Zellen oder Materialien. Im Körper sind zunächst alle Punkte gelistet, wobei ein Punkt pro Zeile mit seinen Eigenschaften notiert wird. Danach werden die Facetten nach einem festen Prinzip referenziert. Zuerst kommt die Anzahl der Punkte pro Fläche, gefolgt von deren Koordinaten und gegebenenfalls noch Eigenschaften.

```
ply
format ascii 1.0          { ascii/binary, format version number }
comment made by Greg Turk { comments keyword specified, like all lines }
comment this file is a cube
element vertex 8          { define "vertex" element, 8 of them in file }
property float x           { vertex contains float "x" coordinate }
property float y           { y coordinate is also a vertex property }
property float z           { z coordinate, too }
element face 6             { there are 6 "face" elements in the file }
property list uchar int vertex_index { "vertex_indices" is a list of ints }
end_header                 { delimits the end of the header }
0 0 0                      { start of vertex list }
0 0 1
0 1 1
0 1 0
1 0 0
1 0 1
1 1 1
1 1 0
4 0 1 2 3                  { start of face list }
4 7 6 5 4
4 0 4 5 1
4 1 5 6 2
4 2 6 7 3
4 3 7 4 0
```

Abbildung 7: Aufbau des PLY-Formats (paulbourke.net, 2013)

Abbildung 7 zeigt ein einfaches Beispiel für den Aufbau einer .ply-Datei. Hier ist sie in ASCII kodiert und besteht aus acht Vertices die in Kombination sechs „faces“ ergeben (paulbourke.net, 2013), (Thies, 2012, S. 29).

3.1.4 Drawing Interchange File Format (DXF)

Das Drawing Interchange File Format ist ein von der Firma Autodesk entwickeltes CAD-Format, das von dem Programm AutoCAD erzeugt wird. Wegen seines leichten Imports und Exports in andere CAD-Programme ist es besonders gut für einen programmübergreifenden Datenaustausch von 3D-Grafiken geeignet. Seit dem Release von AutoCAD wird DXF mit jeder neuen Version erweitert (Weber, 2011, S. 191). Auch dieses Dateiformat kann im ASCII- oder Binärformat erstellt werden.

Der Aufbau des Formates lässt sich in vier Teile gliedern: Kopfteil („Header“), Tabellen („Table“), Blockdefinitionen („Block“) und geometrischer Inhalt der Zeichnung („Entity“). Im Kopfteil werden hauptsächlich allgemeine Informationen über die Zeichnung hinterlegt. So werden in ihm Variablennamen und ihre Werte für die spätere Wiederverwendung festgelegt. Die ebenfalls im Kopfteil abgespeicherten Einstellungen des Quellsystems dienen später im Zielsystem als Voreinstellung. Der Tabellenabschnitt enthält Tabellen mit Definitionen für die Darstellung geometrischer Informationen wie Linientypen, Layer, Textdarstellung, Koordinatensysteme sowie die Darstellung von Bemaßungen oder die Darstellung an Bildschirmen oder ähnlichem (Pülz, 2013, S. 3ff.), (Weber, 2011, S. 192). Der Bereich für Blockdefinitionen dient der Zusammenfassung geometrischer Grundobjekte zu komplexen Objekten. Diese können später beliebig oft als Einzelobjekte referenziert werden. Des Weiteren können auch programmdefinierte Relationen wie Schraffungen oder assoziative Maßketten zusammengefasst werden. Der Entity-Abteil enthält schließlich die geometrischen

Daten zur Zeichnung (Linien, Texte, Polygone, Kreise usw.). Diese werden noch um Attribute wie Farbe, Layer, Strichstärke oder Linientyp ergänzt gespeichert. Überdies werden noch Installationsanweisungen für die im Block-Abschnitt definierten Gruppen gegeben, wodurch Translation, Rotation und Skalierung für die einzelnen Instanzen festgelegt werden (landesvermessung.sachsen.de, 2006, S. 2f.). DXF verwendet Gruppencodes, wie in dem Beispielcode in Abbildung 8 zu sehen ist. Diese werden auf autodesk.com beschrieben: „Beim DXF™-Format werden alle Informationen aus einer AutoCAD-Zeichnungsdatei mit Hilfe spezieller Kennzeichen dargestellt. Die *Kennzeichen* sind Ganzzahlen (so genannte *Gruppencodes*), die jedem Datenelement in der Datei vorangestellt werden. Der Wert der Gruppencodes bestimmt den Typ des nachfolgenden Datenelements.“ (Autodesk.com, 2013). Durch die Anwendung der „Layertechnik“, also die Darstellung der einzelnen Ebenen ist das Format für additive Fertigung geeignet (Grolla, 2005, S. 4 - 9).

```

// header section
0 "SECTION"
2 "HEADER"
...
0 "ENDSEC"
...
// tables section
0 "SECTION"
2 "TABLES"
...
// layer table
0 "TABLE"
2 "LAYER"
70 (# of layers - do not use for reading)
...
// each layer
0 "LAYER"
2 (layer name)
62 (color code)
6 (line style)
...
0 "ENDTAB"
...
0 "ENDSEC"
...
// blocks section
0 "SECTION"
2 "BLOCKS"
...

// each block
0 "BLOCK"
8 (layer name - always 0?)
2 (block name)
70 (flags - always 64?)
10 (base point X)
20 (base point Y)
...
// each entity
0 (entity type)
... (specific for entity)
...
0 "ENDBLK"
...
0 "ENDSEC"
...
// entities (top level cell)
0 "SECTION"
2 "ENTITIES"
...
// each entity
0 (entity type)
... (specific for entity)
...
0 "ENDSEC"
...
// end of file
0 "EOF"

```

Abbildung 8: Aufbau einer DXF-Datei (klayout.de, 2013)

3.1.5 Virtual Reality Modeling Language (VRML)

“VRML ist eine Beschreibungssprache für dreidimensionale Welten bzw. Szenarien, die eine Modellierung von beliebigen dreidimensionalen Objekten ermöglicht, mit denen der Benutzer mit Tastatur und Maus interagieren kann.“ (Schmeil, 2003, S. 2). Aktuell ist die Version 2.0 dieser Sprache, welche den UTF-8 Zeichensatz verwendet und bei dem es sich um eine HTML-Erweiterung handelt. Zur Darstellung wird ein sogenannter „Viewer“ benötigt, welcher entweder ein eigenständiges Programm ist oder als Plug-In zu einem bestehenden Browser als VRML-Browser benutzt wird. Eine VRML-Datei hat die Dateierweiterung „.wrl“ und kann prinzipiell in einem einfachen Editor erstellt werden. In der ersten Zeile werden Version und Zeichensatz als Kommentar beschrieben. Der Aufbau besteht aus einer hierarchischen Auflistung von Objekten in Form von Knoten. Zur Erstellung von Objekten werden vordefinierte Formen (Quader, Kegel, Zylinder, Kugel) mit Default-Werten verwendet. Sollten

die Grundformen nicht ausreichen, können eigene Strukturen zur Wiederverwendung erstellt werden. Der erste Knoten muss immer ein Gruppenknoten sein, der weitere Knoten („children“) enthält. Es gibt fünf verschiedene Gruppenknoten zur Auswahl, welche ihre Eigenschaften an die children vererben. Beispielhaft sei auf den „Transform“-Gruppenknoten zu verweisen, durch welchen Verschiebungen, Rotationen oder Skalierungen erfolgen können. Die Unterknoten sind dann weitere Knoten, wie z.B. der „shape“-Knoten, in welchem die Objekte definiert werden. Den Knoten können Eigenschaften in sogenannten „fields“ übergeben werden. In den „fields“ werden unter anderem die Materialbeschaffenheit, die Farbe oder die Texturen beschrieben. Alle Objekte können beliebig gedreht, gestreckt und kombiniert werden. Um komplexere Figuren zu gestalten, können ähnlich wie beim PLY-Format verschiedene Punkte in ein Koordinatensystem eingegeben und verbunden werden (Klaschke, 2013); auch: (Debacher, 2003).

```
#VRML V2.0 utf8
Cylinder
{
  height 2.0
  radius 2.0
}
Transform
{
  translation 0.0 2.0 0.0
  children
  [
    Cone
    {
      bottomRadius 2.5
    }
  ]
}
```

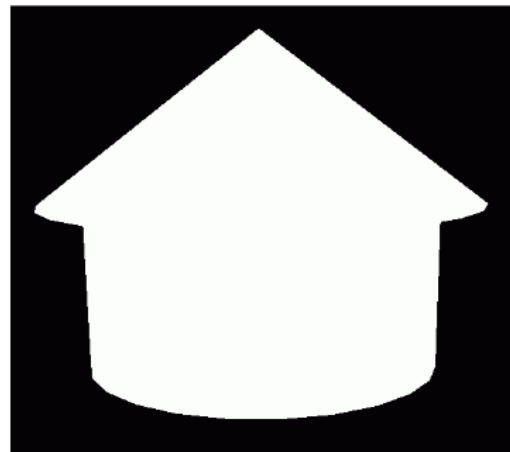


Abbildung 9: Beispiel einer VRML-Datei (Kurth, 2013, S. 8)

Abbildung 9 zeigt eine einfache VRML Datei die einen Zylinder mit einem Kegel kombiniert. Zunächst wird der Zylinder erstellt und in Höhe und Radius angepasst, bevor der Kegel mit verändertem unterem Radius generiert und verschoben wird.

3.1.6 Autodesk 3D Studio (3DS)

Autodesk 3D Studio ist wie auch AutoCAD ein Grafikprogramm von der Firma Autodesk für Modellierung und Animation von 3D-Szenen. Es verwendet das Binärformat mit der Dateiendung „.3ds“ sowie das ASCII-Format mit der Dateiendung „.asc“.

Die Struktur des 3DS Formates ist aus sogenannten „Chunks“ aufgebaut. Dabei handelt es sich um in Blöcke zusammengefasste einzelne Dateielemente, die beliebig in der Datei platziert werden können und so eine flexible Struktur ermöglichen (vgl. Scherfgen 2006, 327). Chunks können selbige enthalten. So beginnt jeder Chunk mit einer „Chunk_ID“, der einen ID-Wert enthält, um die Art der kommenden Daten festzulegen, gefolgt von einer „Chunk_length“, der festlegt, wieweit die folgenden Daten diesem Chunk zugeordnet werden. Danach folgen die

eigentlichen Daten. Chunks lassen sich der Hierarchie nach absteigend zwischen Primär-Chunks („primary“), Haupt-Chunks („main“) und untergeordneten Chunks („subordinate“) unterscheiden (Thierfelder, 2013, S. 21). Die Gestaltung des 3DS-Formates funktioniert mittels Triangulation, wobei die maximale Anzahl an Vertices und Polygonen hier auf 65536 pro Polygonnetz begrenzt ist. Sollte die Anzahl überschritten werden, wird das Objekt aufgeteilt und unter demselben Namen, mit einer zusätzlichen Zahl versehen, gespeichert. Das Format kann keine Normalen übergeben, weswegen sogenannte „smoothing groups“ gebildet werden, eine Gruppierung von aneinander liegenden Polygonen die eine glatte Oberfläche bilden sollen. Über diese „Gruppen“ können dann die Normalen berechnet werden (Wikipedia.de, 2013). Zudem können Texturen und Materialien gespeichert werden (okino.com, 2013).

3.2 Vergleich der Dateiformate

Nachdem ein Überblick über eine Auswahl von vorhandenen Dateiformaten gegeben wurde, sollen diese nun anhand der obigen Beschreibungen verglichen werden.

Das STL-Format ist nach wie vor der de facto Industriestandard und überzeugt durch seinen einfachen Aufbau und seine Akzeptanz bei fast allen 3D-Programmen. Durch seinen einfachen mathematischen Aufbau kann es problemlos an jeder beliebigen z-Koordinate geschnitten werden und ist zugleich noch beliebig skalierbar. Die nachträgliche Bearbeitung kann auch in einem beliebigen externen CAD-Programm stattfinden. Das zur Konstruktion verwendete Programm ist dafür also nicht zwingend erforderlich. Da die Datenelemente mit relativ einfachen Mitteln beschrieben werden, sind Syntaxfehler in der ASCII Version leichter zu erkennen und zu beheben als z.B. bei konturorientierten Schnittstellen. Dies gründet in der höherwertigen Geometrieinformation eines Dreiecks im Gegensatz zu dem Vektor eines Konturzuges. Die großen Schwachstellen des Formates sind der sehr hohe Speicherbedarf und das durch die Triangulation vorhandene Annäherungsproblem. Die bereits beschriebenen Beschreibungsfehler sind STL-typische Mängel und ziehen zumeist eine Nachbearbeitung der STL-Dateien nach sich, damit das erstellte Objekt auch gefertigt werden kann. Das Format bietet keine Möglichkeit technische Daten zu übertragen, das heißt Informationen wie Farbe oder Materialeigenschaften können beim Druck nicht bestimmt werden.

Der Aufbau der Wavefront-Datei ist dem von STL sehr ähnlich, nur dass hier nicht ausschließlich Dreiecke, sondern Polygone ergänzt um Freiform-Objekte verwendet werden. Dadurch können Kurven besser dargestellt werden. Eine Annäherung durch viele kleine Dreiecke entfällt, was eine niedrigere Speicherkapazität zur Folge hat. Dazu kommt, dass durch die Gruppenbildung die Datei zum Großteil redundanzfrei ist. Das Dateiformat ist sehr weit verbreitet und wird fast von allen Grafikprogrammen akzeptiert. In einer OBJ-Datei können per se keine Farbeigenschaften o.ä. abgelegt werden, dafür benötigt es die „mtl“-Datei. Dies ist einerseits günstig, weil sie mit „.obj“ kompatibel ist, andererseits aber umständlich, da immer zwei Dateien übertragen werden müssen und die Folgeinstanz erst auf Kompatibilität überprüft werden muss.

Eine einfache Gestaltung und Redundanzfreiheit lässt sich beim PLY-Format vorfinden. Durch den strukturierten Aufbau und das ASCII-Format ist die Datei sehr übersichtlich, gut lesbar und sie benötigt nur wenig Speicherkapazität. Ein großer Vorteil ist, das Speichern von zusätzlichen Eigenschaften direkt in der Datei. So kann das Objekt farblich gestaltet werden und sogar Bilddateien können referenziert werden. Trotz der Übersichtlichkeit ist die Nachbearbeitung („postprocessing“) nur sehr komplex durchzuführen, da die komplette Datei überarbeitet werden muss und nicht einfach einzelne Abschnitte angepasst werden können. Ebenfalls gegen das Format spricht die Schwierigkeit beim Erstellen von Rundungen. Da das Format auf der Basis von Polygonen erstellt wird, hat man denselben Annäherungseffekt wie bei STL-Dateien. Ein weiteres Defizit ist, dass pro Datei jeweils nur ein Objekt dargestellt werden kann.

Besonders gut geeignet für den Austausch von 3D-Dateien ist das DXF-Dateiformat. Dieses wurde ursprünglich für den Dateiaustausch entwickelt und wird von einem Großteil der CAD-Programme akzeptiert. Durch den Aufbau in einzelne „Layer“ ist es zudem gut geeignet für die additive Fertigung. Es bietet die Möglichkeit verschiedene Eigenschaften in den einzelnen Schichten zu speichern, wodurch es einem Farbdruck zu Gute kommt. Die Möglichkeit der Definition von Gruppenobjekten macht es größtenteils redundanzfrei und somit arbeitsökonomisch für den Ersteller. Ein großer Nachteil ist die große Datenmenge, die beim Erstellen von DXF-Dateien benötigt wird. Damit einher geht ein sehr langsamer Zugriff auf die Datei. Überdies wird oft bemängelt, dass bei der Übertragung zu anderen Programmen viele Ungenauigkeiten entstehen und eine Nachbearbeitung fast unumgänglich ist. Dass mit jeder neuen AutoCAD-Version auch das DXF-Format aktualisiert wird führt oft zu Verwirrungen darüber, welche Version gerade vorliegt.

Tabelle 1: Vergleich der Dateiformate für Volumenmodelle - * Weitergabe in eigener mtl.-Datei

		Dateiformat für Volumenmodelle					
		STL	OBJ	PLY	DXF	VRML	3DS
Kriterium	Triangulierte Darstellung	✓	✓	✓	✓	✓	✓
	Weitergabe von Materialeigenschaften	☒	*	✓	✓	✓	✓
	redundanzfrei	☒	✓	✓	✓	✓	☒
	Einfache Erweiterbarkeit	☒	✓	✓	✓	✓	✓
	Geringe Speichergröße	☒	✓	✓	☒	☒	✓
	Weitergabe von Farbinformationen	☒	*	✓	✓	✓	✓
	ASCII - Kodierung	✓	✓	✓	✓	✓	☒ (.asc)
	Binärkodierung	✓	☒	✓	☒	☒	✓

			(.mod)		(.dxb)		
Verbessertes Slicing	✓	✓	✓	☒	✓	✓	☒
Hoher Verbreitungsgrad	✓	✓	✓	✓	✓	✓	✓
Beliebige Skalierung	✓	✓	✓	✓	✓	✓	☒
Normalenvektordarstellung	✓	✓	✓	✓	✓	✓	☒
Gute Übertragung zu anderen Programmen	✓	✓	✓	☒	✓	☒	✓
Verwendung von Freiform-Kurven	☒	✓	✓	☒	✓	☒	☒
Einfache Nachbearbeitung	✓	✓	✓	☒	✓	✓	✓

VRML-Daten werden stets als ASCII-Files gespeichert, wodurch sie sich zwar sehr leicht erzeugen und bearbeiten lassen aber auch sehr viel Speicherplatz benötigen. Je detaillierter und komplexer die Objekte werden desto immenser werden die Datenmengen. Bei einer hohen Anzahl von modellierten Objekten wird die Datenmenge nur noch schwer handhabbar. Ein Nachteil der Browserabhängigkeit besteht darin, dass verschiedene Plug-Ins dieselbe Datei unterschiedlich darstellen und die Datei erst nach komplettem Download sichtbar ist. Für das Dateiformat sprechen die einfache Handhabung und die Möglichkeit der Übergabe von zusätzlichen Eigenschaften, sodass mit VRML farbige Objekte gedruckt werden können.

Die 3DS-Dateien sind durch ihren „Chunk“-Aufbau einfach zu ergänzen oder zu bearbeiten. Es können Textur und Farbeigenschaften übergeben werden. Da das Format binär kodiert ist, ist es für den Menschen nicht lesbar, was durch mangelnde Informationsbereitstellung von Autodesk noch verstärkt wird. Bei der Fertigung kann es zu Problemen kommen, da die Normalen nicht übergeben werden. Diese werden zwar durch „smoothing groups“ berechnet, hierbei gibt es aber keine Garantie auf Genauigkeit.

Um einen guten Überblick über die Verwendbarkeit der einzelnen Dateiformate zu geben, wurden in Tabelle 1 prägnante Eigenschaften, so wie die Vor- und Nachteile der einzelnen Formate zusammengefasst.

4 Datentransformation beim Slicen

Sobald das Modell als 3D-Datei vorhanden ist, muss es in einzelne Schichten zerteilt werden. Dieser Vorgang nennt sich „Slicen“ und wird entweder mittels Triangulation im STL-Format vollzogen oder direkt am Bauteil geschnitten (SLI- oder CLI-Format). Je nach zu fertigender Anlage kann ein spezielles Dateiformat benötigt werden, wobei das STL-Format in aller Regel jedoch akzeptiert wird.

4.1 SLI-/CLI-Format

Die CLI- (SLI-, SLC-)Schnittstelle, auch konturorientierte Schnittstelle genannt, enthält die Geometriedaten der einzelnen zu fertigenden Schichten eines Bauteils. Die Datenübergabe bei der CLI-Schnittstelle erfolgt auf eine systemübergreifende und anlagenneutrale Art, was sie zwischen einzelnen Fertigungssystemen austauschbar macht.

Die SLI- oder SLC- File hingegen überträgt auch anlagenspezifische Zusatzdaten was einen Austausch zwischen verschiedenen Anlagen verhindert (Gebhardt, 2007, S. 28). Die Erzeugung der Schnittebenen findet direkt im CAD-Programm, genau genommen in der Schicht statt. Die Informationen reichen hierbei aus, um neben der Kontur auch Schraffierungen angeben zu können, welche die Kontur vollständig beschreiben kann. Alle Formulierungen welche die Kontur und Schraffierung beschreiben, werden in der Höhenkoordinate gespeichert um ein Schneiden am mathematisch exakten Objekt zu ermöglichen. Bezüglich der Querschnittsfläche wird zwischen einer inneren und einer äußeren Berandung und Schraffierung unterschieden (O.V., 2013, S. 5).

Durch die isolierten Schichtinformationen und den fehlenden Bezug zum Verlauf der Höhenkoordinate ist eine nachträgliche Bearbeitung der Skalierbarkeit nicht möglich. Die Datei muss, anders als bei einer STL-Datei entweder im Original-CAD bearbeitet oder neu angelegt werden. CLI-Files enthalten die Schnittkonturen von allen Bauebenen, wobei eine eindeutige Definition von Außen- und Innenkonturen elementar ist um eine fehlerfreie Produktion zu gewährleisten. Sollten diese fehlerhaft sein, kann es zu einem Abbruch oder einer fehlerhaften Produktion kommen. Beispielsweise bei der Stereolithographie können fehlerhaften Innen- oder Außenkonturen den Aufbau eines inneren Volumens verursachen, obwohl das Bauteil als hohl geplant war oder das Gegenteil bewirken, also den Aufbau eines Volumens verhindern (Gebhardt, 2007, S. 29f.).

4.2 Weitere Schnittstellenformate (STL, NSF, RPDF)

Das STL-Format ist eine Standardschnittstelle von CAD-Programmen zu Fertigungsanlagen, es wird in der Regel immer akzeptiert. Durch die Beschreibung der gesamten Geometrie durch Dreiecke, ist jeder Dreieckspunkt auf der Randkontur mathematisch eindeutig beschrieben. Sollte beim Schneiden in die einzelnen Schichten, eine Dreieckskontur durchgeschnitten

werden, so kann der Schnittpunkt auf der Verbindungslinie zweier Punkte eindeutig beschrieben werden (Westkämpfer, 2006, S. 148). Darüber hinaus existieren noch zwei weitere Schnittstellenformate, NSF und RPDF. NSF ist ein an der Universität Stuttgart entwickeltes Format, welches einer besseren Prozessplanung dienen und wesentliche Ansätze zur integrierten Übertragung von Geometrie und Technologie enthalten soll. Das Rapid Prototyping Data Format (RPDF) wurde am Fraunhofer Institut entwickelt und besitzt laut Dreher das größte Potential hinsichtlich der Übertragung technologischer Daten (Dreher, 2005, S. 22f.).

4.3 Vergleich der Schnittstellenformate

Alle Formate sind gut lesbar, redundanzfrei, konvertierbar und erweiterbar. Außerdem können diese die vollständige Geometrie des eigentlichen Objektes gut bis sehr gut widerspiegeln. Große Unterschiede gibt es bei der Weitergabe von technologischen Informationen, dem Informationsgehalt, der Formulierung und der Darstellung von Segmenten.

Tabelle 2: Vergleich der Dateiformate für Schichtmodelle (Dreher, 2005, S. 23)

Eigenschaften	CLI	SLC	NSF	RPDF
Vollständigkeit Geometrie	→	↗	↗	↑
Vollständigkeit Technologie	↓	↓	→	↑
Darstellung von Segmenten	↓	↓	↑	↑
Neutrale Formulierung	↘	↘	↗	↗
Exaktheit	→ (Polylinien)	→ (Polylinien)	→ (Polylinien)	↗
Vermeidung von Redundanz	↗	↗	↗	↗
Erweiterbarkeit	→	→	↗	↗
Konvertierbarkeit	↗	↗	→	→
Informationsgehalt	↘	↘	↗	↗
Lesbarkeit	↗	↗	↗	↗

In Tabelle 2 werden die Schnittstellenformate anhand wesentlicher Kriterien miteinander verglichen, wobei das STL-Format aufgrund seines Mangels technische Daten zu übertragen, außen vor gelassen wird.

5 Ausblick auf zukünftige Dateiformate

Das STL-Format bildet den Industriestandard im Bereich der CAD-Modellierung und der additiven Fertigung, ist jedoch schon seit über drei Jahrzehnten im Einsatz. Deshalb soll dieses Format in Zukunft durch das neue Additive Manufacturing File Format (AMF) abgelöst werden. Ein weiteres Format, welches zukünftig vermehrt Anwendung im 3D-Druck finden wird, ist das erst kürzlich von Microsoft vorgestellte 3D Manufacturing Format (3MF).

5.1 Additive Manufacturing File Format (AMF)

AMF arbeitet wie STL mit einem Netz aus Dreiecken, jedoch können noch zusätzlich gebogene Kanten an den Dreiecksflächen verwendet werden. Auch kann das auf XML basierende Dateiformat zusätzliche Informationen wie Bauparameter, Material oder Farbe übergeben (Breuninger, 2013, S. 57). Der Aufbau ist sehr simpel gestaltet, redundanzfrei und in ASCII kodiert. STL-Dateien können ohne Datenverlust und ohne Angabe zusätzlicher Informationen in AMF umgewandelt werden und auch eine Abwärtskompatibilität ist gewährleistet, wobei hier die erweiterten Fähigkeiten des AMF-Formates verloren gehen. Damit es auch auf zukünftigen Systemen anwendbar ist, kann es beliebig erweitert werden. Für eine geringere Dateigröße sorgt die Möglichkeit eines komprimierten Speicherns, wie in Tabelle 3 gezeigt.

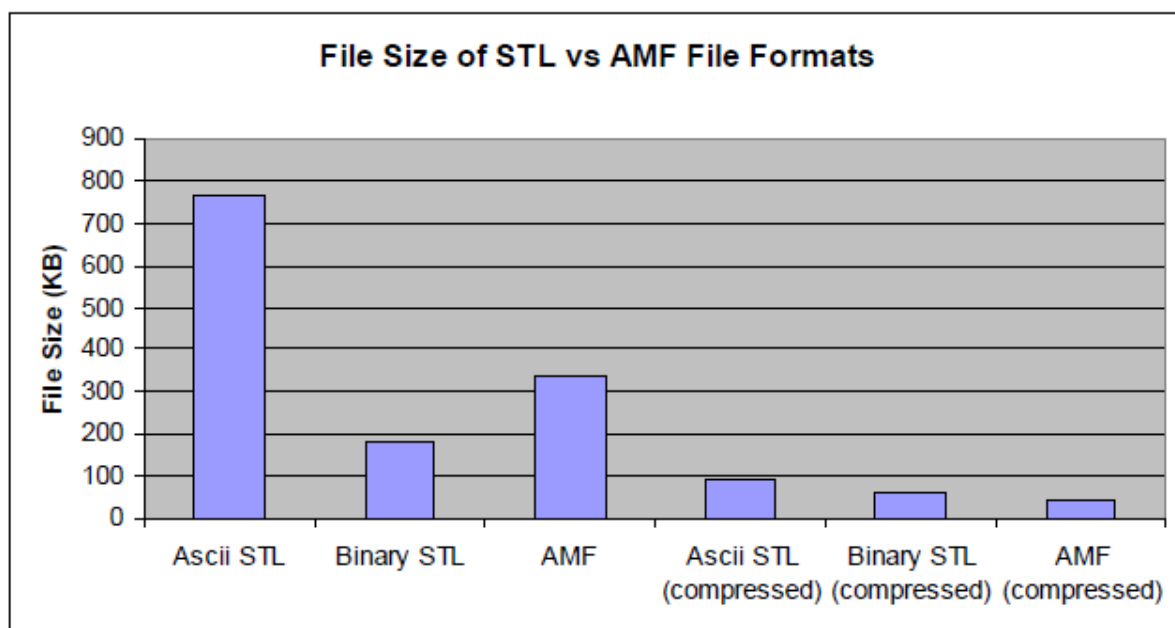


Abbildung 10: Vergleich Dateigröße AMF-STL (Hiller, 2013, S. 11)

Der Aufbau des AMF-Formates kann im Internet nachgelesen werden (Creativemachines.cornell.edu, 2013, S. 2 - 5). Hier wird auch die Weitergabe von Farbinformationen und Krümmungen erklärt.

AMF wurde bereits vor einem Jahr als Nachfolger von STL angekündigt. Jedoch hat sich das Format bis heute nicht durchgesetzt, was hauptsächlich an der fehlenden Implementierung vieler großer Software- und Anlagen-Hersteller geschuldet ist. (3Druck.com, 2013).

5.2 3D Manufacturing Format (3MF)

Im Rahmen der Einführung von Windows 8.1 wurde von Microsoft ein neues 3D Manufacturing Format (3MF) vorgestellt. Die Besonderheit liegt darin, dass Windows 8.1 das erste Betriebssystem ist, welches umfassenden integrierten Support für den 3D-Druck bietet. Ziel von Microsoft ist dabei eine hochwertige Umgebung zu schaffen, die den 3D-Druck einerseits deutlich benutzerfreundlicher macht und andererseits einen Anreiz schafft, sich dem Thema über Apps im Windows Store zu nähern. 3MF wurde entwickelt, weil die aktuell gängigen Formate die Erwartungen von Microsoft für eine Einbettung in das Betriebssystem nicht befriedigen konnten. Über den Aufbau des Formates existieren noch keine Details. Bisher ist nur bekannt, dass es auf XML basiert und Farben-, sowie Materialeigenschaften unterstützt. (Gear, 2013).

6 Zusammenfassung und Ausblick

Im der vorliegenden Arbeitsbericht wurde der allgemeine Entstehungsprozess eines Objekts bei der additiven Fertigung dargestellt, wobei zwischen CAD-Modellierung und 3D-Scan unterschieden wurde. Während der 3D-Scan gute Ansätze aufweist, ist wohl auch in Zukunft eine Modellierung in einem CAD-Programm nicht vermeidbar, da die Detailliertheit, gerade was das Innere eines Objektes angeht, durch das Erfassen mit bestehenden Scan-Methoden nicht in dem Maße erreicht werden kann, wie sie bei einer Vielzahl der Anwendungen benötigt wird.

In dem Hauptteil des Berichts, welcher den „Vergleich von Dateiformaten für 3D-Modelle“ behandelte, wurden einzelne Dateiformate vorgestellt und miteinander verglichen. Dabei wurde deutlich, dass es zwar eine große Anzahl an Dateiformaten gibt, keines davon aber alle Eigenschaften aufweist, um in Zukunft alles Nötige bereitzustellen, um den 3D-Druck großflächig abzudecken. So existiert zwar das STL-Format, welches sich über die Jahre als Industriestandard durchgesetzt hat; aber gerade aufgrund der mangelnden Übertragung von Farb- und Materialeigenschaften ist es nicht für jedes Objekt anwendbar. Der Aufbau der verschiedenen Dateiformate ähnelt sich dabei sehr und lässt sich zumeist nur anhand des benötigten Speicherbedarfs und der Möglichkeit von zusätzlichen Elementen, wie beispielsweise Rundungen, unterscheiden. Die beiden Autodesk Formate DXF und 3DS werden zwar weiterhin ihren Anteil auf dem Markt haben, sind aber Autodesk-spezifisch und daher nicht für den universellen Heimbedarf geeignet.

Die schichtweise Fertigung hat sich durchgesetzt und erzielt gute Ergebnisse. Gegen sie spricht jedoch der Umstand, dass die Dateien vor der Fertigung zusätzlich noch auf die in Kapitel 4 beschriebenen Formate übertragen werden müssen.

Die einfache Handhabung und die Unterstützung durch das weit verbreitete Betriebssystem Windows 8.1 machen die beiden Formate AMF und 3MF kommerziell erfolgsversprechend. AMF bringt vieles mit, was ein Dateiformat benötigt: Einen einfachen Aufbau, die Möglichkeit, sämtliche Strukturen wiederzugeben sowie die Übermittlung von Farb- und Materialeigenschaften. Dennoch wirkt 3MF mit Blick in die Zukunft am bedeutendsten, gerade durch die Unterstützung von Windows 8.1. Einen 3D-Druck am heimischen PC mit einem eigenen 3D-Drucker leicht durchzuführen zu können, ohne auf Spezialkenntnisse angewiesen zu sein, ist auf lange Sicht erstrebenswert. Allerdings ist es noch zu früh 3MF als neuen Standard zu bezeichnen, da über das Format noch zu wenig Details bekannt sind.

Literaturverzeichnis

- 3Druck.com. (2013). *Dateiformate: .AMF das neue STL? (Update)*. Von <http://3druck.com/programme/dateiformate-amf-das-neue-stl-305703/>, zugegriffen am 29.09.2013 abgerufen
- Autodesk.com. (2013). *DXF-Format (DXF)*. Von <http://docs.autodesk.com/ACD/2013/DEU/index.html?url=files/GUID-3F0380A5-1C15-464D-BC66-2C5F094BCFB9.htm,topicNumber=d30e663204>, zugegriffen am 04.10.2103 abgerufen
- Breuninger, J. e. (2013). *Generative Fertigung mit Kunststoffen- Konzeption und Konstruktion für selektives Lasersintern*. Berlin: Springer-Verlag.
- Creativemachines.cornell.edu. (2013). *Standard Specification für Additive Manufacturing File Format (AMF)*. Von http://creativemachines.cornell.edu/sites/default/files/AMF_V0.47.pdf, zugegriffen am 01.10.2013 abgerufen
- Danjou, s., & Koehler, H. (2013). *Vorbereitung von CAD-Konstruktiondaten für den RP-Einsatz - Eine Schnittstellenproblematik*. Von <http://www.rtejournal.de/ausgabe5/1509>, zugegriffen am 22.09.2013 abgerufen
- Debacher, U. (2003). *VRML-Einführung*. Von <http://www.debacher.de/vrml/vrml.htm>, zugegriffen am 24.09.2013 abgerufen
- Diemke, J. (2010). *Wavefront OBJ Format*. Von http://www.informatik.uni-oldenburg.de/~trigger/content/opengl/opengl_course/slides/2010-JOGL-10-Wavefront-OBJ.pdf, zugegriffen am 20.09.2013 abgerufen
- Dreher, S. (2005). *Flexible Integration von Rapid Prototyping Prozessketten in die Produktentstehung*. Berlin: Fraunhofer IPK Verlag.
- Fastermann, P. (2012). *3D-Druck/Rapid Prototyping*. Berlin: Springer-Verlag.
- Fasterpoly. (2011). *So wird in 3D gedruckt, eine Beschreibung des Verfahrens*. Von http://www.fasterpoly.de/de_wp/2011/04/was-ist-3d-druck/, zugegriffen am 23.09.2013 abgerufen
- Friedrich, R. (2012). *Merkmale Generativer Fertigungsverfahren*. Von http://www.mp.haw-hamburg.de/fachberg/prod/vorlesungen_ptm/RP/RPT_3_2012_HAW.pdf, zugegriffen am 25.09.2013 abgerufen
- Gear, G. (2013). *3D Printing Support in Windows 8.1 explained*. Von <http://blogs.windows.com/windows/b/extremewindows/archive/2013/08/22/3d-printing-support-in-windows-8-1-explained.aspx>, zugegriffen am 01.10.2103 abgerufen

- Gebhardt, A. (2007). *Generative Fertigungsverfahren: Rapid Prototyping – Rapid Tooling – Rapid Manufacturing*. 3. Aufl. München: Carl Hanser-Verlag.
- Grolla, S. (2005). *Interpretieren von DXF-Dateien*. Von [http://www.uni-weimar.de/cms/fileadmin/bauing/files/professuren/bauinf/lehre/CIB1/Belege/Grolla Beleg.pdf](http://www.uni-weimar.de/cms/fileadmin/bauing/files/professuren/bauinf/lehre/CIB1/Belege/Grolla%20Beleg.pdf), zugegriffen am 24.09.2013 abgerufen
- Hiller, J. (2013). *Draft Additive Manufacturing File ormat*. Von amf.wikispaces.com/file/view/AM2010_lipson.ppt/237127968/AM2010_lipson.ppt, zugegriffen am 01.10.2103 abgerufen
- Klare, M., & Gischer, F. (2009). *Grundlagen des Rapid Prototyping für Dental Applikationen*. Von http://www.ddn-online.net/uploads/smartsection/501_ddn_1109_klare_2.pdf, zugegriffen am 19.09.2013 abgerufen
- Klaschke, G. (2013). *Einführung in VRML 2.0*. Von http://mume.fh-friedberg.de:6060/sirius/M_Lutz_GG2/Html/VRML/VRMLIntro.htm, zugegriffen am 24.09.2013 abgerufen
- klayout.de. (2013). *DXF Format*. Von http://www.klayout.de/dxf_format.html, zugegriffen am 30.09.2013 abgerufen
- Kurth, W. (2013). *Grundzüge von VRML*. Von http://www.uni-forst.gwdg.de/~wkurth/cg08_v12.pdf, zugegriffen am 24.09.2013 abgerufen
- landesvermessung.sachsen.de. (2006). *Kurzbeschreibung der DXF-Schnittstelle*. Von http://www.landesvermessung.sachsen.de/inhalt/produkte/lika/alk/download/Schnittstelle_DXF_0603.pdf, zugegriffen am 25.09.2013 abgerufen
- O.V. (2013). *Babylon II*. Von http://www.fh-dortmund.de/de/fb/5/forschung/cpsim/medien/pdfs/Babylon_II_S.pdf, zugegriffen am 28.09.2013 abgerufen
- okino.com. (2013). *Okino's 3d Studio.3ds Exporter*. Von http://www.okino.com/conv/exp_3ds.htm, zugegriffen am 04.10.2013 abgerufen
- paulbourke.net. (2013). *PLY - Polygon File Format*. Von <http://paulbourke.net/dataformats/ply/>, zugegriffen am 22.09.2013 abgerufen
- Pülz, G. (2013). *Analyse des „Formblatt zum DXF –Datenaustausch“ des DBV*. Von <http://puelz.net/dxf.pdf>, zugegriffen am 23.09.2013 abgerufen
- Reddy, M. (2013). *B1. Object Files (.obj)*. Von <http://www.martinreddy.net/gfx/3d/OBJ.spec>, zugegriffen am 25.09.2013 abgerufen
- Scherfgen, D. (2006). *3D-Spiele-Programmierung*. München: Hanser-Verlag.
- Schmeil, A. (2003). *3D-Grafik und -Animation - VRML - Virtual Reality Modelling Language*. Von <http://www-vs.informatik.uni->

- ulm.de/teach/ws03/vp/ausarbeitungen/VRML_Schmeil.pdf, zugegriffen am 21.09.2013 abgerufen
- Schwanecke, U. (2010). *Übungsblatt 1*. Von <http://www.mi.hs-rm.de/~schwan/Vorlesungen/InCG/Uebungen/blatt1.pdf>, zugegriffen am 29.09.2013 abgerufen
- Thierfelder, K. (2013). *Dateiformate für dreidimensionale Daten*. Von <http://cs.uni-muenster.de/u/lammers/EDU/ws03/Landminen/Abgaben/Gruppe7a/Thema07a-3DDatenformate-KoljaThierfelder.pdf>, zugegriffen am 04.10.2013 abgerufen
- Thies, M. (2012). *Web-basierte Visualisierung von Mobile-Mapping Daten*. Von http://www.ikg.uni-hannover.de/fileadmin/ikg/staff/thesis/finished/documents/MA_Thies.pdf, zugegriffen am 22.09.2013 abgerufen
- Weber, N. (2011). *Facettenbasierte Indexierung multipler Artefakte - ein Framework für vage Anfragen in der Produktentwicklung*. Bamberg: University of Bamberg Press.
- Westkämpfer, E. (2006). *Einführung in die Organisation der Produktion*. Berlin: Springer-Verlag.
- Wikipedia.de. (2013). *Smoothing Group*. Von http://en.wikipedia.org/wiki/Smoothing_group, zugegriffen am 04.10.2013 abgerufen

Über die Autoren



Heiko Heckner studiert im sechsten Semester Wirtschaftsinformatik an der Universität Würzburg und hat vor kurzem seine Bachelorthesis zum Thema „Einsatzbereiche und Nutzenpotenziale von In-Memory Data-Management“ verfasst. Im Moment absolviert er ein sechsmonatiges Praktikum bei der Softwareentwicklungsfirma „Trinodis“ in Würzburg.



Marco Wirth, M.Sc. studierte Informatik mit Nebenfach Wirtschaftswissenschaften an den Universitäten Paderborn und Helsinki. Seit 2012 ist er als wissenschaftlicher Mitarbeiter am Lehrstuhl für Wirtschaftsinformatik und Systementwicklung der Universität Würzburg tätig. Im Rahmen seines Engagements im Kompetenzzentrum CEDIFA beschäftigt er sich mit Fragestellungen rund um das Gebiet der additiven Fertigung.

Kontakt

Lehrstuhl für Wirtschaftsinformatik und Systementwicklung
Julius-Maximilians-Universität Würzburg
Josef-Stangl-Platz 2
D-97070 Würzburg

T +49 (0)931 3180242
F +49 (0)931 3181268
E kontakt@cedifa.de
W <http://www.cedifa.de>