

AD3D – Eine automatisierte Toolchain für das 3D-Drucken

Stefan Kupper¹, Verena Mühlig¹, Laura Jakobeschky¹

Zusammenfassung

Der technische Fortschritt hat dazu geführt, dass die Kosten für das automatisierte Drucken von Emblemen, Anhängern sowie allgemein gedruckten Prototypen stark gefallen sind. Dies geschah insbesondere auch vor dem Hintergrund eines Wandels der Industrie hin zu einer modernen Industrie 4.0. In dieser Arbeit wird ein neuer Ansatz für das automatische 3D-Drucken eines Logos präsentiert, welches als Pixelgrafik vorliegt. Die vorgeschlagene Prozesskette erzeugt hierbei final den G-code zum Drucken. Dabei sind seitens des Users wenige Eingaben erforderlich.

Stichwörter

Additive Fertigung, 3D-Druck, Industrie 4.0, Marketing



Logo des Unternehmens	Bilderdaten-transfer in CAD	3D-Daten mit Logo	Slicer	3D-Druck	Nachbearbeitung
Beliebige einfarbige Bilddatei kann gewählt werden	Bilddaten werden extrahiert und auf CAD-Modell aufgetragen	STL-Datei mit projiziertem Logo wird exportiert	Druckparameter werden je nach Filament festgelegt	Bauteil wird additiv gefertigt	Entfernung von überschüssigem Material

Abbildung 1: Schematische Darstellung der Verarbeitungskette für die automatische Generierung eines gedruckten 3D-Logos mit der vorgeschlagenen Toolchain. Der beispielhaft gezeigte Anwendungsfall der Erstellung eines Schlüsselanhängers dient der unmittelbaren Veranschaulichung. Der in dieser Arbeit vorgestellte Prozess ist allerdings allgemein anwendbar.

1 Einleitung

Der 3D-Druck kann aktuell bereits als ein etabliertes Verfahren gelten, welches ein breites Anwendungsfeld gefunden hat [1, 2]. In der Medizin ist die Umsetzung von 3D-Bilddaten in einen 3D-Druck bereits seit längerer Zeit in der praktischen Umsetzung gegeben [3, 4]. Es existieren nach Kenntnis der Autoren jedoch keine Ansätze, um zweidimensionale Abbildungen umsetzen zu können. In verschiedenen Kontexten von Industrie 4.0 bis hin zu Konferenzen kann es jedoch bedeutsam werden ein Logo, welches als Pixelgrafik vorliegt, physisch im 3D-Druck erstellen zu können. In dieser Arbeit geht es deshalb darum, einen solchen Modellansatz vorzustellen, um mithilfe dieses Ansatzes prinzipiell eine automatisierte additive Fertigung umsetzen zu können. Damit kann die Automatisierung bestehender Systeme fokussiert und vorangetrieben werden.

¹ Technische Hochschule Wildau, Hochschulring 1, 15745 Wildau

Der Ablauf der von uns entwickelten Prozesskette lässt sich schematisch dabei wie in Abbildung 1 darstellen. Ausgehend von einer Logodatei wird sequentiell ein Dateiumwandlung bzw. ein Dateitransfer durchgeführt, um schlussendlich das gedruckte Logo erhalten zu können. Im folgenden werden wir den sequentiellen Prozess – ausgehend anhand vom eigentlichen Prozessansatzes – detailliert darlegen.



Abbildung 2: Beispiele für Logos die in dieser Arbeit zur demonstrativen Veranschaulichung genutzt werden. *linke Bildhälfte:* das Logo für das Mittelstand-Digital Zentrum Spreeland *rechte Bildhälfte:* das Logo der Technischen Hochschule Wildau. Beide Logos haben (in Bezug auf die additive Fertigung) eine unterschiedliche geometrische Komplexität und erfordern ggf. eine Nachbehandlung durch den Nutzer. Im konkreten Beispielfall muss im Falle des linken Logos von Hand eingegriffen werden, während das rechts gezeigte Logo diese Eingriffe nicht erfordert.

2 Prozessansatz

Um den Prozess gut abbilden bzw. erstellen zu können, ist es wichtig sich zunächst grundlegende Überlegungen zu machen, was wesentliche geometrische Eigenschaften des finalen 3D-Objektes betrifft. Es handelt sich bei den vorgeschlagenen automatisch erstellten 3D-Objekten um quasi-prismatische Figuren. Das bedeutet, dass das 2D-Logo in ein quasi-Prisma übertragen wird, wobei erhöhte Bereiche dem eigentlichen Logo entsprechen. Das entstandene Element, welches das Logo enthält, kann nun auf einen geeigneten Basis-Körper platziert werden. Theoretisch ist dieser frei konfigurierbar, praktisch unterliegt der Basis-Körper physikalisch material-bedingten Grenzen. In dieser Beschreibung wird sich deshalb zunächst auf einen simplen rechteckig-prismatischen Hohl- oder Festkörper beschränkt. Ein Hohlkörper geht dabei mit einer Materialersparnis einher, dafür aber je nach Material auch mit einem gewissen Stabilitätsverlust. Grundsätzlich kann der zugrundeliegende Basiskörper dabei frei konfiguriert. Dies betrifft dabei Form, Farbe und Material. In dieser Arbeit werden wir den Prozess exemplarisch anhand der von uns gewählten Materialparameter beschreiben.

Mithilfe des zuvor beschriebenen prinzipiellen Ablaufes bzw. Vorgehens ist die entstandene geometrische Figur eindeutig im dreidimensionalen Raum definiert. Im folgenden Abschnitt werden wir deshalb die Prozesskette des Modelles beschreiben.

3 Prozesskette des Modelles

3.1 Verarbeitung der Logo-Datei

Um den eindeutig definierten dreidimensionalen Körper drucken zu können, ist eine sequentielle Umwandlung der Bildinformationen des Logo-Bildes in einen entsprechenden G-code notwendig. Zu diesem Zweck wird das Logo zunächst in ein Binärbild umgewandelt. Dabei können wir uns zunutze machen, dass das Bild eines Logos klar und rauschfrei definiert ist (siehe Referenzbeispiele aus Abbildung 2). Dies ermöglicht es, die Kanten des Logos mit einem Canny-Edge-Detektor zu ermitteln. Zur Anwendung dieses Verfahrens haben wir die frei verfügbare Bibliothek OpenCV genutzt [5]. Der

Canny-Edge-Detektor zur Erkennung von Kanten wiederum ist ein bereits seit Jahren etabliertes numerisches Verfahren [6] welches es ermöglicht, die Bildkoordinaten der Randpunkte auszulesen. In dieser Arbeit und für dieses Modell konnten wir entsprechend auf den in OpenCV implementierten [7] Detektor zurückgreifen.

Es ist an dieser Stelle nützlich, auf eine mögliche Problemquelle, welche relativ spezifisch für Logo-Abbildungen ist, hinzuweisen. Es ist möglich, dass der Kanal für die Transparenz des Bildes als 4-ter Kanal nicht einheitlich definiert ist. Im Resultat kann es passieren, dass Strukturen im eingelesenen Bild im Hintergrund erscheinen, die nicht zum eigentlichen Logo gehören. Das liegt daran, dass typischerweise in der Darstellung und Interpretation dieser Bilder über reguläre Darstellungsmethoden für die Bilder wie im einfachen Bildbetrachter der 4-te Kanal komplett ignoriert wird und das Problem somit nicht offensichtlich wird. Für unsere Prozesskette sorgt ein vorgeschalteter selbst geschriebener Konverter dafür, dass diese Artefakte automatisch aus dem Bild vor der Weiterverarbeitung entfernt werden.

Wendet man nun den Canny-Edge-Detektor an, so werden unmittelbar am Rand des Bildes liegende Elemente des Logos nicht erkannt. Wie in ähnlichen Problemfeldern [8] lässt sich dies jedoch über eine einfache Schwellwert Erkennung vor dem Hintergrund detektieren und das Binärbild kann um diese Punkte ergänzt werden. Abbildung 3 zeigt nun die entstandenen Binärbilder die in der Prozesskette weiterverarbeitet werden können.



Abbildung 3: Resultierende Binärbilder berechnet aus den Beispiellogos aus Abbildung 2. *linke Bildhälfte*: das zum Logo für das Mittelstand-Digital Zentrum Spreeland korrespondierende Binärbild *rechte Bildhälfte*: das zum Logo der Technischen Hochschule Wildau korrespondierende Binärbild.

3.2 Skript-basierte verbleibende Prozesskette

Das resultierende Binärbild lässt sich nun mithilfe eines geeigneten Programmes vektorisieren, d. h. in ein Vektorformat überführen. Für die Zwecke der Demonstration der prinzipiellen Machbarkeit wird in dieser Arbeit das bekannte Open Source Programm Inkscape genutzt (<https://inkscape.org/>). Der große Vorteil hierbei ist die Ansteuerung des Programmes über die Kommandozeile, so dass sich eine Skript-basierte Ausgestaltung der Prozesskette ermöglichen lässt. Konkret wird die Umwandlung des resultierenden Binärbildes in das Vektorformat über die Kommandozeile initiiert. Im nächsten Schritt werden die Punkte aus der Vektorgrafik auf den Basiskörper virtuell integriert. Dabei lässt sich das Open Source Programm OpenSCAD nutzen (<https://openscad.org/>) welches skriptbasiert arbeiten kann. Es lässt sich eine STL-Datei extrahieren, welche zur Ermittlung des G-codes genutzt werden kann. Dieser finale Schritt, d. h. die Erstellung des G-codes mit dem Slicer lässt sich nicht generalisiert im Skript-Prozess abbilden, da die Vielfalt der verwendeten Programme und der Maschinen die final zur additiven Fertigung benutzt werden zu groß ist. Es ist jedoch einsichtig, dass für eine gegebene Hardware die für die additive Fertigung genutzt wird, eine vollautomatisierte Umsetzung machbar ist.

4 Sinnvolle Anpassungsparameter

Natürlich lässt sich prinzipiell eine enorme Bandbreite der Konfigurierbarkeit ermöglichen, schon allein wegen einer enormen Varianz an möglichen Basiskörpern. Aufgrund praktischer Beschränkungen und zur Illustration beschränken wir uns in dieser Arbeit auf die relevantesten Konfigurationsmöglichkeiten. Dies sind die Bereite der Konturlinie auf dem Basiskörper, die Höhe des Logos über dem Basiskörper, die Höhe des Basiskörpers selber und die Breite des Basiskörpers. Aus rein ästhetischen und materialbedingten Gründen lässt sich allen Größen für jeden Anwendungsfall eine typische Skale zuordnen. In dieser Arbeit beziehen wir uns im Hinblick auf konkrete Parameterwerte auf die Anwendung der Erstellung eines Schlüsselanhängers. Typische Parameterwerte für diesen Anwendungsfall sind in Tabelle 1 dargestellt. Die Parameter lassen sich den Skripts in Form von Konfigurationsdateien zugrunde legen und erlauben so eine individuelle Konfiguration.

Tabelle 1: Übersicht über wichtige Parameter in der Prozesskette. Gezeigt sind typische Werte für den beispielhaften Anwendungsfall eines Schlüsselanhängers.

Parameter	Referenzwert	Effektiver Parameterbereich
Breite Einzelkontur	90 (% Düsendurchmesser)	85–120 (% Düsendurchmesser)
Höhe des Logos über Basiskörper	1 mm	0,5–2 mm
Höhe Basiskörper	~ 5 mm	5–10 mm
Breite Basiskörper	~ 20 mm	20–40 mm

5 Schlussfolgerungen

Wir konnten in dieser Arbeit aufzeigen, wie man eine Prozesskette erstellen kann, welche in der Lage ist grundsätzlich automatisch aus einer gegebenen Logodatei einen 3D-Druck zu erstellen. Dieser 3D-Druck ist dann über geeignete Parameter (siehe Tabelle 1 für eine Übersicht) konfigurierbar und gestattet somit eine flexible Nutzung einer solchen Prozesskette.

Der wesentliche Vorteil der von uns vorgestellten Prozesskette liegen darin, dass die Erstellung der individuellen Druckdaten zeitlich und räumlich getrennt von der eigentlichen additiven Fertigung erfolgen kann. Insbesondere sind die in dieser Arbeit beschriebenen Elemente der Prozesskette grundsätzlich und weitgehend digitalisierbar. Damit ließe sich alles beschriebene grundsätzlich in einer Server-Client Architektur umsetzen.

Wie wir in dieser Arbeit zeigen konnten, lassen sich leider nicht alle Logos vollautomatisiert (aufgrund geometrischer Einschränkungen) umsetzen. So ist das linke Logo aus Abbildung 2 nicht unmittelbar geeignet und muss manuell nachbearbeitet werden, was daran liegt, dass das automatische Anwählen geschlossener Kurven in diesem Falle nicht gelingt. Im Vergleich dazu ist das entsprechende rechte Beispiellogo unmittelbar umsetzbar. Nichtsdestotrotz leistet die in dieser Arbeit vorgestellte automatische Toolchain eine effiziente und nützliche Vorarbeit für den Prozess der additiven Fertigung. Die Autoren halten den Ansatz deswegen auch grundsätzlich für geeignet, um modellhaft einen kleinen Beitrag zum Übergang der Industrie 3.0 hin zur Industrie 4.0 ermöglichen zu können. Wesentliche Bestandteile des Codes mit entsprechender Dokumentation werden auf der Webseite der Arbeitsgruppe zur Verfügung gestellt werden und sind somit online abrufbar: <https://www.th-wildau.de/forschung-transfer/forschung/ic3/>.

6 Literatur

- [1] Jordan, J. 3D Printing. (MIT Press, 2019)
- [2] Evans, B. Practical 3D printers: The science and art of 3D printing. (Apress, 2012)
- [3] Rengier, F., Mehndiratta, A., Von Tengg-Kobligk, H., Zechmann, C., Unterhinninghofen, R., Kauczor, H. & Giesel, F. 3D printing based on imaging data: review of medical applications. *International Journal Of Computer Assisted Radiology And Surgery*. 5 pp. 335–341 (2010)
- [4] Mitsouras, D., Liacouras, P., Imanzadeh, A., Giannopoulos, A., Cai, T., Kumamaru, K., George, E., Wake, N., Caterson, E., Pomahac, B. & Others Medical 3D printing for the radiologist. *Radiographics*. **35**, 1965–1988 (2015)
- [5] Bradski, G. The openCV library.. *Dr. Dobb's Journal: Software Tools For The Professional Programmer*. **25**, 120–123 (2000)
- [6] Rong, W., Li, Z., Zhang, W. & Sun, L. An improved CANNY edge detection algorithm. *2014 IEEE International Conference On Mechatronics And Automation*. pp. 577–582 (2014)
- [7] Xu, Z., Baojie, X. & Guoxin, W. Canny edge detection based on Open CV. *2017 13th IEEE International Conference On Electronic Measurement & Instruments (ICEMI)*. pp. 53–56 (2017)
- [8] Komarudin, A., Satria, A. & Atmadja, W. Designing license plate identification through digital images with OpenCV. *Procedia Computer Science*. **59** pp. 468–472 (2015)