

3D-Druck zur haptischen Darstellung von Grafiken und Graphen in der Blinden- und Sehbehindertenpädagogik

Jonathan Hofmann¹, Thorsten Thormählen¹, Knut Büttner²

Philipps-Universität Marburg, AG Grafik und Multimedia Programmierung¹

Carl-Strehl-Schule, Deutsche Blindenstudienanstalt e.V. Marburg²

k.buettner@blista.de

Zusammenfassung

Die steigende Popularität und die damit verbundene bessere Verfügbarkeit von 3D-Druckern bietet viele Chancen im Bereich der Blinden- und Sehbehindertenpädagogik. Die Nachfrage der Schüler nach tastbaren Objekten ist hoch und 3D-Drucker kommen in Schulen für Sehbehinderte vermehrt zum Einsatz.

Dieser Beitrag beschreibt, wie der 3D-Drucker auch abseits des typischen, voluminöseren 3D-Drucks zum Erstellen von flacheren, taktilen Grafiken und Graphen (2.5-dimensionalen Objekten) genutzt werden kann.

Hierfür wurde eine barrierefreie Webanwendung entwickelt, die es blinden oder sehbehinderten Benutzern erlaubt, einfache 2D-Grafiken und mathematische Graphen mit Braillebeschriftungen zu erstellen. Diese vektorbasierten 2D-Grafiken werden in von einem 3D-Drucker ausführbaren G-Code konvertiert und in mehreren 2D-Ebenen schnell und effizient auf einen passenden Druckträger gedruckt.

1 Einleitung

In der Blinden- und Sehbehindertenpädagogik existiert ein hoher Bedarf an taktilen Unterrichtsmaterialien, von der bekannten Blindenschrift Braille bis hin zu haptischen Repräsentationen von Graphen und Grafiken. Der Prozess, diese herzustellen, ist je nach verwendetem Verfahren mit mehr Aufwand verbunden als das einfache Ausdrucken eines Blatt Papiers. Die zunehmende Popularität der 3D-Drucker und die immer weiter fallenden Preise dieser Geräte, machen sie auch für den Unterricht mit sehbehinderten Menschen interessant. Meist beschränkt sich die Verwendung des 3D-Drucks jedoch auf das klassische Drucken von voluminösen, dreidimensionalen Objekten.

Veröffentlicht durch die Gesellschaft für Informatik e. V. 2018 in

R. Dachse, G. Weber (Hrsg.):

Mensch und Computer 2018 – Tagungsband, 02.–05. September 2018, Dresden.

Copyright (C) 2018 bei den Autoren. <https://doi.org/10.18420/muc2018-mci-0345>

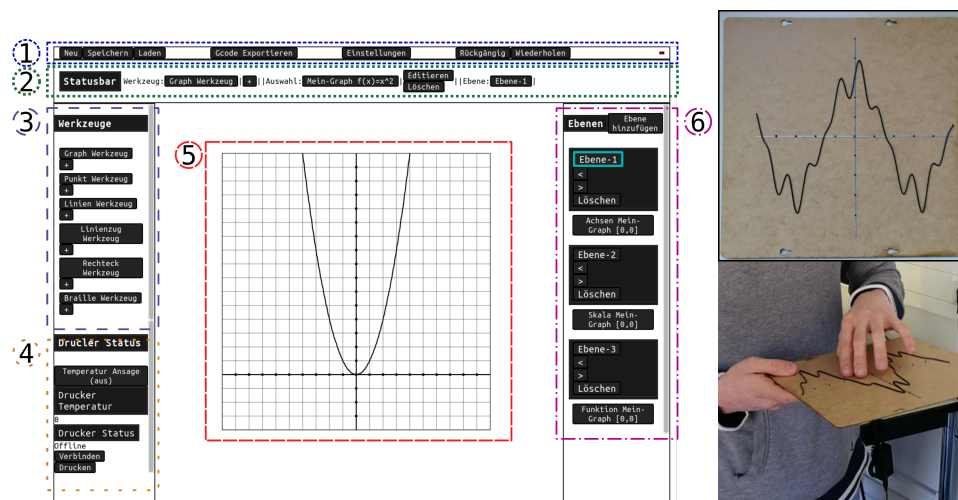


Abbildung 1: Links: Überblick über die Webanwendung. (1) Menüleiste, (2) Statusbar; (3) Werkzeuge, (4) Drucker Status, (5) Zeichenfläche, (6) Ebenen; Rechts oben: Gedruckter 2D-Graph; Rechts unten: Schüler ertastet den Graphen

Dies inspirierte die Frage, ob nicht die besagten Graphen und Texte zuverlässig und schnell auch mit handelsüblichen 3D-Druckern gedruckt werden können, um etwa einen schnellen Zugriff auf Unterrichtsmaterialien zu gewährleisten oder um blinden und sehbehinderten Benutzern selbst die Möglichkeit zu geben, diese zu erstellen und auszudrucken. Daher wurde eine barrierefreie Webanwendung entwickelt, mit der Benutzer einfache, vektorbasierte Grafiken erstellen und mathematische Graphen plotten können, um diese mit einem 3D-Drucker auf einen passenden Druckträger, als haptisch wahrnehmbare Grafik, zu drucken.

Beispiel für eine Arbeit mit ähnlichen Zielen ist SVGPlott (Bornschein et al., 2014). Dies ist jedoch eine Desktop-Anwendung und auf das Drucken auf Schwellpapier ausgelegt. In Viz-Touch (Brown und Hurst, 2012) werden ebenfalls ertastbare mathematische Funktionsgraphen mit dem 3D-Drucker erstellt. Im Gegensatz zu diesem Ansatz wird in diesem Beitrag auf einen Druckträger, wie stärkeres Papier oder dünnes MDF-Holz, gedruckt, was Druckmaterial und Zeitaufwand reduziert. In Linespace (Swaminathan et al., 2016) wird in ähnlicher Weise auf eine große Tafel gedruckt. Allerdings ist der Druck dort auf nur eine aufgebraute 2D-Ebene beschränkt.

2 Barrierefreies webbasiertes 2D-Zeichenprogramm

Die Webanwendung verwendet die WAI-ARIA-Spezifikationen als Erweiterung der HTML-Semantik, mit der sich recht unkompliziert eine für Screenreader gut lesbare, barrierefreie Webseite gestalten lässt.

Die Oberfläche ist in mehrere Werkzeugleisten gegliedert, die sich um die Zeichenfläche gruppieren. Für eine barrierefreie Navigation ist es möglich, die einzelnen Werkzeugleisten mit

der Tabulatortaste zu durchwandern. Die Reihenfolge entspricht der Nummerierung in Abbildung 1, wobei die Zeichenfläche ausgelassen wird. Bis auf die oberste Menüleiste, welche einfache Dateioperationen bereitstellt, sind die Werkzeugleisten jeweils mit einer Überschrift versehen, die vor allem sehbehinderten Benutzern die Information zur Verfügung stellt, wo genau innerhalb der Anwendung sie sich gerade befinden. Gefolgt wird diese Überschrift dann von den eigentlichen Funktionen der Leiste. Die mit den Funktionen belegten Buttons werden, wo es nötig ist, durch ein entsprechendes ARIA-labeled-by Attribut unterstützt, um deren Funktionsweise in textueller Form für Screenreader besser zu verdeutlichen.

Die barrierefreie Navigation innerhalb der Anwendung ist ein wichtiger Schwerpunkt der vorliegenden Arbeit. Nach verschiedenen Versuchen haben sich zwei recht einfache Abläufe herauskristallisiert. Zum einen bieten viele Screenreader die Möglichkeit, Überschriften durchzublättern. Somit war es naheliegend, die Werkzeugleisten jeweils mit einem Überschrift-HTML-Tag zu versehen. Die andere Möglichkeit, schnell zwischen den Werkzeugleisten hin und her zu navigieren, bietet ein per einfachem Shortcut erreichbares Navigations-Menü.

Der Schwerpunkt der Anwendung liegt auf dem Erstellen von Grafik-Objekten, wie etwa einem Linienzug oder auch komplexeren Objekten, wie Graphen. Zum einen ist es für sehende Benutzer möglich, über die Steuerung per Mausklick direkt auf dem Zeichenfeld Grafikobjekte zu erstellen. Da dies jedoch nicht den Schwerpunkt der Anwendung darstellt, wurde auch die Möglichkeit geschaffen, Objekte textuell zu erzeugen und zu editieren. Um das barrierefreie Arbeiten mit grafischen Objekten zu ermöglichen, kann jedes Objekt mit Hilfe eines formularbasierten Dialogfensters hinzugefügt oder später modifiziert werden. Die Formulare beinhalten vor allem Angaben zu Koordinaten oder Dimension, mit denen die einzelnen Grafikobjekte beschrieben werden. Neben den weiteren objektspezifischen Einstellungen, existieren auch Felder, um die Objekte zu benennen und damit dem Benutzer die Möglichkeit zu bieten, diese in der Ebenenleiste wieder zu finden.

Um die Steuerung des 3D Druckers für blinde und sehbehinderte Nutzer zugänglicher zu machen wurde diese mit Hilfe eines OctoPrint-Servers direkt in die Anwendung integriert.

3 3D-Druck von 2D-Grafikobjekten

Inzwischen existiert eine Auswahl an Programmen, sogenannte Slicer¹, um 3D-Objekte in für den 3D-Drucker lesbaren G-Code umzuwandeln. Jedoch stellte sich heraus, dass diese nicht besonders geeignet sind, um schnell einfache Linienzüge oder gar Punkte zu drucken. Somit musste eine eigene Konvertierung von den in der Applikation erstellten 2D-Objekten in G-Code geschrieben werden.

Viele G-Code Kommandos sind von ihrer Grundstruktur her einer 2D-Vektorbeschreibung so ähnlich, dass eine einfache Umrechnung der Vektorposition in Millimeter auf dem tatsächlichen Druckbereich den Großteil des Codes vorgibt und nur Parameter, wie die Extrusionsmenge oder die Geschwindigkeit hinzugefügt werden müssen. Beispielsweise besteht der Druckbefehl für eine einfache Linie (nachdem der Druckkopf zum Startpunkt des Linie bewegt wurde) aus dem

¹beispielsweise [Ultimaker-Cura](#) oder [Slic3r](#)

G1-Kommando gefolgt von den X-, Y- und Z-Koordinaten des Endpunktes der Linie in mm. Der E-Parameter gibt an, wieviel Filament vom Drucker eingezogen werden soll und der F-Parameter bezeichnet die Maximalgeschwindigkeit in mm/min.

```
G1 X20.0 Y20.0 Z0.6 E3.0 F3000
```

Eine besondere Herausforderung stellte der Druck von einzelnen Punkten dar, welcher für Braillezeichen oder auch Achsenskalierungen genutzt werden. Zunächst wird die Basis des Punktes auf halber Ebenenhöhe gedruckt. Dies geschieht mit einer Bewegung einzig und allein auf der Z-Achse. Danach wird diese Basis in noch warmen und weichen Zustand einen Zehntel Millimeter mit dem Druckkopf nach unten gedrückt. Dadurch bildet sich eine gute Basis für den eigentlichen Punkt und die Haltbarkeit auf dem Trägermaterial wird erhöht. Im zweiten, darauffolgenden Schritt wird nun die obere Hälfte des Punktes gedruckt, um die gewünschte Ebenenhöhe zu erreichen.

Die korrekte Berechnung des Extrusions-Parameters ist ein kritischer Bestandteil des Druckes, denn wenn zu viel Filament ausgegeben wird, ist das Druckbild unscharf oder der Druckkopf zieht Fäden hinter sich her. Daher muss berechnet werden, wie viel Filament der Drucker einziehen muss, um eine Linie der Breite w , der Höhe h und Länge l auszugeben. Außerdem hängt die Berechnung vom Durchmesser d des Filaments ab. Diese Berechnung entspricht einer einfachen Volumenberechnung, wobei der Querschnitt der Linie nicht durch ein einfaches Rechteck, sondern durch ein Rechteck mit abgerundeten Ecken approximiert wird:

$$E = \frac{l \left(\pi \left(\frac{h}{2} \right)^2 + h(w - h) \right)}{\pi \left(\frac{d}{2} \right)^2}$$

Um haptisch besser unterscheidbare Objekte zu generieren, können die Objekte schon in der Anwendung in Ebenen aufgeteilt werden, die dann verschiedenen Druckhöhen entsprechen. Die Ebenen entscheiden auch über die Reihenfolge des Druckes. So muss, um eine Kollision mit bereits gedruckten Objekten zu vermeiden, zunächst die unterste Ebene gedruckt werden, inklusive aller ersten Schichten der darüber liegenden Ebenen. So bauen sich die weiteren Ebenen inkrementell in 0,6mm hohen Schichten auf. Dieser inkrementelle Aufbau der höheren Objekte bietet den Vorteil, dass der G-Code zweier Ebenen sich nicht voneinander unterscheidet. Einzig der Z-Parameter muss entsprechend der Anzahl an Ebenen angepasst werden.

4 Ergebnisse erster Praxisversuche mit Schülern

Die Webanwendung wurde mehreren blinden Schülern zur Verfügung gestellt. Diese konnten ohne größere Probleme eigenständig Graphen erstellen und in 3D ausdrucken. Ein Schüler benutzte die Anwendung um eine im Unterricht verwendete, trigonometrische Funktion zu modifizieren und dabei herausfinden, wie sich der Graph dieser neuen Funktion im Vergleich zu der ursprünglichen verhält (siehe Abb. 1). Damit konnte ihm eine neuartige Möglichkeit zur selbstständigen Beschäftigung mit der mathematischen Thematik geboten werden.

Literaturverzeichnis

- Bornschein, J., Prescher, D. & Weber, G. (2014). SVGPlott – Generating Adaptive and Accessible Audio-Tactile Function Graphs. In *Computers Helping People with Special Needs, International Conference on Computers for Handicapped Persons* (S. 588–595).
- Brown, C. & Hurst, A. (2012). VizTouch: Automatically Generated Tactile Visualizations of Coordinate Spaces. In *Proc. of International Conference on Tangible, Embedded and Embodied Interaction* (S. 131–138).
- Swaminathan, S., Roumen, T., Kovacs, R., Stangl, D., Mueller, S. & Baudisch, P. (2016). Linesspace: A Sensemaking Platform for the Blind. In *Proc. of Conference on Human Factors in Computing Systems* (S. 2175–2185).