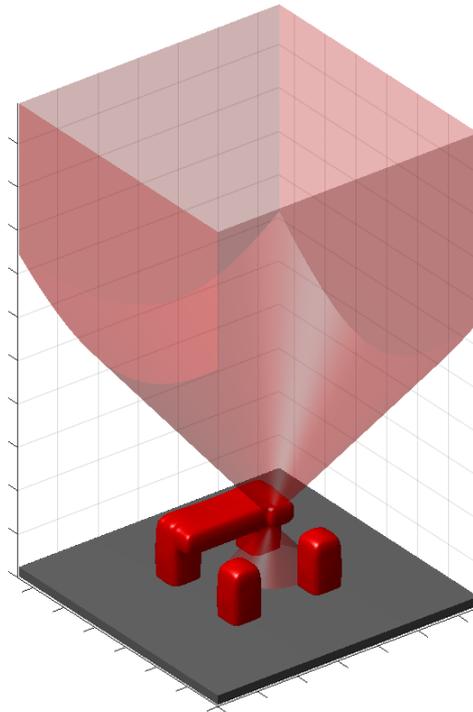


## 3D-Lasernanodruck-Simulation



Abschlussdokumentation der Kooperationsphase 2021/22

Durchgeführt am Institut für Angewandte Physik  
Betreut durch Pascal Kiefer und Tobias Messer

Gestrich, Elias  
Kurs KA 16

Vierling, Dan  
Kurs KA 16

## Abstract

In the past years, the unprecedentedly high resolutions of 3D laser nano lithography created new opportunities in a wide variety of application fields. This technique is based on special resins, which are induced by light to polymerize. Therefore, strongly focused laser beams can be used to create arbitrary three-dimensional nano structures within them. However, since those resins exhibit memory-like behaviour, the printed structures may form unintended shapes. The aim of this project was to visualize this effect.

# Inhaltsverzeichnis

<b>1. Motivation</b>	<b>1</b>
<b>2. Theoretische Grundlagen</b>	<b>2</b>
2.1. Aufbau eines 3D-Laserdruckers . . . . .	2
2.2. Radikalische Polymerisation . . . . .	2
2.3. Quenching . . . . .	4
2.4. Accumulation-Threshold-Model . . . . .	5
2.5. Photonen-Absorptions-Prozesse . . . . .	5
<b>3. Methodik</b>	<b>7</b>
3.1. Modellierung des Lichtfokus . . . . .	7
3.2. Entwicklung des Programms . . . . .	8
3.2.1. Berechnung der Akkumulation . . . . .	8
3.2.2. Eingabe der 3D-Struktur . . . . .	9
3.2.3. Akkumulation . . . . .	10
3.2.4. Kameraeinstellung . . . . .	11
<b>4. Ergebnisse</b>	<b>12</b>
<b>5. Diskussion</b>	<b>14</b>
5.1. Auflösung . . . . .	14
5.2. Laufzeit . . . . .	15
<b>6. Ausblick</b>	<b>16</b>
<b>A. Anhang</b>	<b>17</b>

## 1. Motivation

3D-Drucker gibt es heutzutage in den unterschiedlichsten Ausführungen. Mithilfe dieser können z. B. Bauteile, die konventionell nicht herzustellen wären, individuell angefertigt werden. Dabei werden sie sowohl in Dimensionen von mehreren Metern verwendet, um beispielsweise mit Betondruck Bestandteile für Häuser zu drucken, als auch in handlichen Größen, um Ersatzteile für Haushaltsgegenstände herzustellen. Dies sind alles Beispiele des 3D-Drucks mittels Filamentdruckverfahrens, bei dem flüssiges Material extrudiert wird, welches danach aushärtet. Dabei wird das Material Schicht für Schicht aufgetragen.

In unserem Fall beschäftigen wir uns jedoch mit einer Technologie, die Stereolithographie genannt wird und das Drucken in noch kleineren Maßstäben ermöglicht - im Nanometer-Bereich. Bei dieser Technologie wird durch die Fokussierung eines energiereichen Laser auf einen kleinen Bereich ein spezieller Lack gehärtet. Dadurch können 1000-mal höhere Auflösung realisiert werden, als es die Materialextruder im eben erklärten Filamentdruckverfahren zulassen. Der 3D-Lasernanodruck findet schon heute Anwendung, zum Beispiel bei der Herstellung winziger optischer Linsen und diffraktiver Gitter, und eröffnete das Feld der 3D-Metamaterialien. Diese weisen neuartige Eigenschaften auf, wie z. B. negative Kompressibilitätsfaktoren. Außerdem verspricht die Präzision des Druckverfahrens viel im Hinblick auf medizinische Anwendungen. Angedacht sind hier Gerüste, mit denen Zellen zu gezieltem Gewebewachstum angeleitet werden können. [4]

Verschiedene Lacke absorbieren die eintreffenden Photonen des Lasers auf unterschiedliche Weise. Dabei gibt es zwei Gruppen, in die die bisherigen Lacke unterteilt werden können. Lacke der einen Sorte können zwar mit weniger Laserleistung betrieben werden, sind jedoch für die Komplexität, die die gedruckten Objekte in der Anwendung haben, zu unpräzise. Ziel unseres Projektes war es, das Auftreten dieser Unterschiede zu erklären und mit Simulationen zu veranschaulichen.

## 2. Theoretische Grundlagen

### 2.1. Aufbau eines 3D-Laserdruckers

Alle 3D-Lasernanodrucker, die am Institut für angewandte Physik des KIT verwendet werden, funktionieren nach dem Prinzip der Stereolithographie.

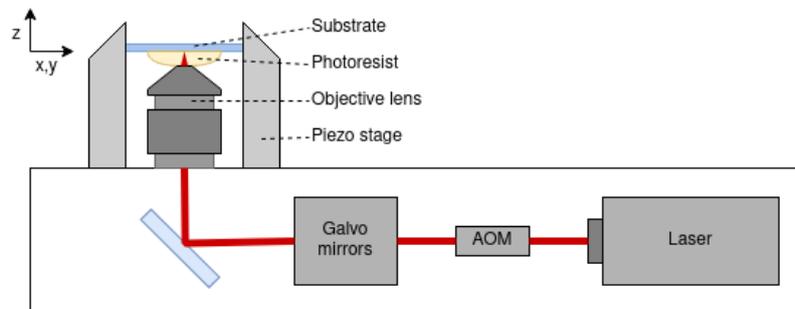


Abbildung 1: Schematischer Aufbau eines 3D-Lasernanodruckers; Eigene Abbildung nach [3]

Als Photonenquellen dienen Femtosekunden-Pulslaser. Sie können Lichtimpulse um die 100 fs erzeugen und erreichen damit Spitzenleistungen über 1 GW. Mit zwei Galvo-Spiegeln, die sich je um eine Achse drehen können, kann der Fokuspunkt in der x-y-Ebene mit konstanten Geschwindigkeiten von bis zu  $0,2 \frac{\text{m}}{\text{s}}$  bewegt werden. Ein akustooptischer Modulator unterbricht zuvor den Laserstrahl zwischen den Schreibvorgängen, während die Spiegel gebremst, neu positioniert und wieder beschleunigt werden müssen. Fokussiert wird der Strahl durch ein Objektiv mit bis zu 100-facher Auflösung, das in direktem Kontakt mit dem Lack steht. Die Entfernung zur Substratplatte kann mit Piezomotoren eingestellt werden. Damit wird der Fokuspunkt in z-Richtung bewegt.

Zu Beginn eines Druckprozesses liegt der Brennpunkt in der Ebene der Substratplatte. Wie bei herkömmlichen Verfahren werden die 3D-Objekte schichtweise gedruckt. Der Abstand, den die Schichten zueinander haben, heißt Slicing. Jede Schicht wiederum wird aus parallelen Linien zusammengesetzt. Der Abstand zwischen zwei Linien heißt Hatching. Wenn der Druckvorgang beendet ist, kann der noch flüssige Lack abgespült werden, sodass die fertige Struktur auf der Substratplatte zurückbleibt. [3]

### 2.2. Radikalische Polymerisation

Die Lacke bestehen im Wesentlichen aus Kunststoffmonomeren und photosensitiven Initiatormolekülen. Die Monomere teilen den selben Grundaufbau (s. Abbildung 2) und zeichnen sich dadurch aus, dass sie mindestens eine Ethenylgruppe besitzen. An dieser werden sie untereinander zu Polymerketten verknüpft. Der Rest R ist von Werkstoff zu Werkstoff verschieden.

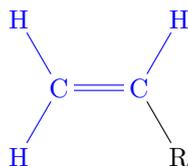


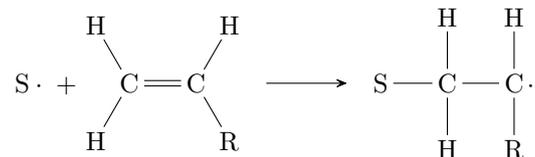
Abbildung 2: Grundaufbau eines Kunststoffpolymers, die Ethenylgruppe blau hervorgehoben

Die Initiatoren sind notwendig, um die Polymerisation anzustoßen. Durch eintreffende Photonen

werden sie in einen energiereicheren Zustand versetzt und können in Folge dessen zu zwei Starterradikalen gespalten werden. Ausgehend davon läuft die Polymerisation in drei Phasen ab.

### Initiierung

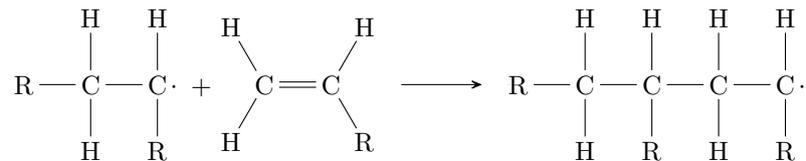
Bei der Initiierung greift ein Starterradikale S die Doppelbindung der Ethenylgruppen eines Moleküls an und löst dabei ein bindendes Elektronenpaar auf. Es bindet kovalent an eines der Kohlenstoffatome.



Das ungepaarte Elektron erscheint am anderen Ende des Monomers und bildet wieder ein Radikal.

### Propagation

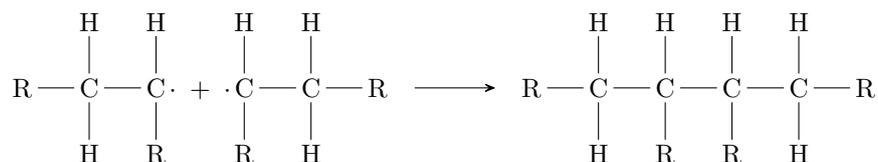
In einer Wachstumsreaktion können sich an diesem Radikal immer weitere Monomere anlagern. Durch den selben Mechanismus wie bei der Initiierung wird das eine Kohlenstoffatom des Monomers gebunden, während das andere ein neues Radikal ausbildet.



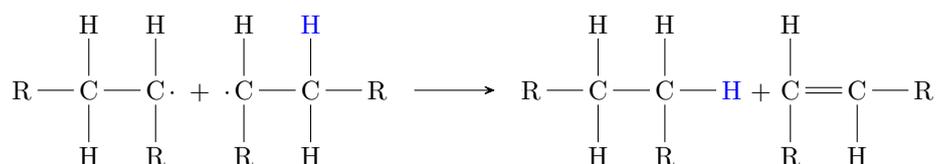
### Termination

Wenn zwei Radikale aufeinander treffen, kommt es zur Abbruchreaktion. Dies kann entweder dadurch geschehen, dass sich zwei Ketten an ihren radikalischen Enden zu einer Kette rekombinieren, oder dass durch die Übertragung eines H-Atoms zwei Ketten mit nicht-radikalischen Enden entstehen.

#### Rekombination



#### Disproportionierung



Anstelle der Polymerradikale können auch andere, z. B. noch vorhandene Starterradikale, an der Rekombination und der Disproportionierung beteiligt sein. Lediglich bei Letzterer muss mindestens einer der beiden Reaktionspartner eine Polymerkette mit radikalischem Ende sein. [5]

Die Termination ist erst bei fortgeschrittener Polymerisation zu beobachten. In Bereichen, in denen die Monomere noch vorwiegend ungebunden vorliegen, trifft ein radikalischer Kettenende mit höherer Wahrscheinlichkeit auf ein Monomer. Dort ist die Propagation wahrscheinlicher. Daher läuft die Polymerisation, einmal initiiert, in einem ausschließlich aus Monomeren bestehenden Lack soweit ungehindert ab, bis sich überall ein Polymernetzwerk gebildet hat. Der Lack ist dann auf makroskopischer Ebene ausgehärtet.

Um dem Kunststoff zusätzliche Stabilität zu verleihen, werden Monomergemischen sogenannte Vernetzungsmittel zugegeben. Das sind Moleküle mit mehreren Ethenylgruppen. Sie nehmen an der Propagation mehrerer Polymerketten teil und bilden somit Verknüpfungen zwischen diesen. Beim 3D-Lasernanodruck sind in der Regel ausschließlich Vernetzungsmittel als Monomeranteil in den Lacken vorhanden. Ein häufig verwendetes Molekül ist Pentaerythritoltriacrylat, kurz PETA. Dessen drei Ethylgruppen sind in Abbildung 3 blau markiert. [2]

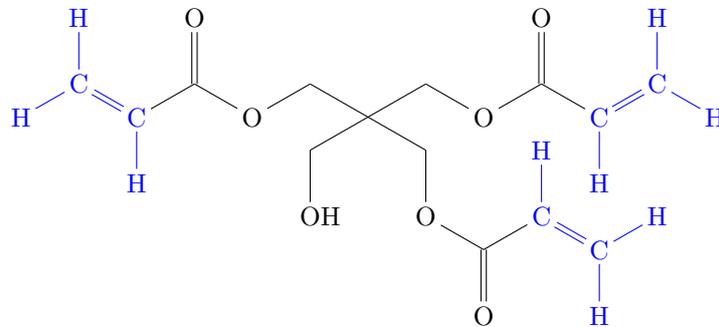


Abbildung 3: Pentaerythritoltriacrylat (PETA)

### 2.3. Quenching

Beim 3D-Lasernanodruck soll verhindert werden, dass der gesamte Lack nach der Initiierung aushärtet. Dafür spielt der Sauerstoff, der aus der Luft in den Lack diffundiert, eine entscheidende Rolle. Er hat die Fähigkeit, die Polymerisation zu unterbinden. Dies nennt man Quenching.

Indem der Sauerstoff Peroxylradikale bildet, erschwert er die Propagation von Polymerketten. Die neu gebildeten Radikale sind stabiler und nehmen selten an einer Wachstumsreaktion teil. Durch

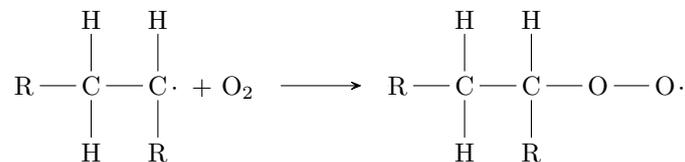


Abbildung 4: Bildung eines Peroxylradikals

Rekombination oder Disproportionierung können sie aber zur Termination führen, wenn sie mit radikalischen Kettenenden aufeinandertreffen. Dadurch wird ein Fortsetzen der Polymerisation unterbunden, wenn keine neuen Starterradikale durch Einwirkung des Lasers gebildet werden. [2]

## 2.4. Accumulation-Threshold-Model

Ein Aushärten an einer Stelle im Lack wird dann erreicht, wenn es zu ausreichend großer Kettenlänge sowie ausreichend vielen Verknüpfungen zwischen den Ketten gekommen ist. Zur Quantifizierung dieses Verhaltens wird die Lichtdosis verwendet. Sie beschreibt, wie viele Initiatormoleküle pro Volumen angeregt werden und hängt direkt von der Lichtintensität ab. Dadurch lässt sich ein Schwellenwert definieren, bei dem der Lack, wenn er einmal dieser Dosis ausgesetzt wurde, aushärtet. [1]

Innerhalb des Strahlenfokus gibt es einen Bereich, in dem der Lack nach einmaliger Einwirkung aushärtet. Die Dauer der Einwirkung ergibt sich dabei aus der Schreibgeschwindigkeit. Angelehnt an das Wort Pixel, das die kleinsten Bildelemente benennt, heißt diese Volumeneinheit Voxel.

Beim 3D-Lasernanodruck kommt es jedoch nicht nur innerhalb dieser Voxel zur Aushärtung des Lacks. Auch in den Ausläufern des gebündelten Strahls, in denen die Dosis unter der Schwelle liegt, kommt es durch einzelne Polymerisationsreaktionen zur Bildung von Polymerketten. Da diese zu einem späteren Zeitpunkt mit anderen Ketten zu größeren Netzen verknüpft werden können, spielt die Gesamtmenge der über die Zeit hinweg deponierten Lichtdosis eine Rolle. Durch Effekte wie der Diffusion ändert sich ihre Verteilung im Raum, als Näherung kann dies jedoch vernachlässigt und stattdessen angenommen werden, dass die Dosis für jeden Punkt im Raum linear aufsummiert wird. Die Ausdehnung der Bereiche, in denen der Lack unbeabsichtigt polymerisiert, hängt dabei stark von der Weise ab, wie die Initiatoren vom einfallenden Licht angeregt werden. [2]

## 2.5. Photonen-Absorptions-Prozesse

Nach Albert Einstein kann ein einzelnes Photon, welches die passende Energie liefert, ein Molekül anregen, wenn diese aufeinander treffen. Dieser Prozess heißt 1-Photon-Absorption (1PA). In einem Lichtbündel trifft aber nicht jedes Photon auch auf ein Molekül. Das Anregen ist ein statistisches Ereignis, was dazu führt, dass die Dosisrate proportional zur Lichtintensität  $I$  ist.

Alternativ zur 1-Photon-Absorption entdeckte Marie Göppert-Mayer die 2-Photonen-Absorption (2PA). Hierbei haben die Photonen nur die halbe Energie. Trifft ein Photon auf ein Initiatormolekül, wird es auf ein virtuelles Energieniveau gehoben. Praktisch zeitgleich muss ein zweites Photon das Molekül treffen, um es in den angeregten Zustand zu versetzen. Die Wahrscheinlichkeit für dieses Ereignis ist geringer, da sie im Vergleich zur 1PA quadriert ist. Die Beziehung zwischen Lichtdosisrate  $\dot{D}$  und Lichtintensität  $I$  ändert sich daher zu  $\dot{D} \sim I^2$ . Der Exponent der Intensität, der bei der 1PA gleich 1 und bei der 2PA gleich 2 ist, wird als Nichtlinearitätswert bezeichnet.

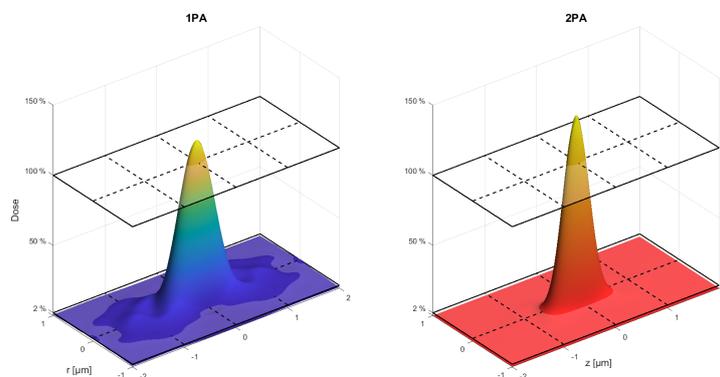


Abbildung 5: Vergleich der Lichtdosis bei 1PA und 2PA; Eigene Abbildung

Durch Justierung der Laserintensität lässt sich die Lichtdosisrate so einstellen, dass die Voxel bei 1PA und 2PA gleich sind. Das heißt, der Bereich, in dem die Dosis nach einmaliger Einwirkung des Lasers über der Schwellendosis liegt, hat beide Male die gleiche Größe. In Abbildung 5 ist das Profil jener Dosis über der optischen Achse  $z$  und dem radialen Abstand  $r$  zur Achse dargestellt.

Der entscheidende Unterschied liegt in der Umgebung des Voxels. Dazu betrachte man die 2%-Höhenlinie bei der 1PA. Diese entspricht der 0,04%-Höhenlinie beim 2PA-Verfahren. Wirkt der Laser 50 Mal an einer Stelle im Lack ein, akkumuliert sich die Dosis innerhalb des abgebildeten Bereiches auf über 100%, wenn das 1PA-Verfahren verwendet wird. Verwendet man dagegen 2PA, muss der Laser 2500 Mal an der Stelle einwirken, damit der ausgehärtete Bereich die gleiche Ausdehnung hat. Bei nur 50-maligem Einwirken ist das Volumen, wie in der Abbildung zu sehen, sehr viel geringer. Dieser Effekt ist für den 3D-Lasernanodruck unentbehrlich, da die Unterschiede schon beim Druck kleiner Strukturen stark ausgeprägt sind. [4]

### 3. Methodik

#### 3.1. Modellierung des Lichtfokus

Die deponierte Lichtdosis im Raum ist durch die Intensität  $I$  des Lasers bestimmt. Im Verlauf der Arbeit wurden zur Modellierung des Intensitätsfeldes zwei verschiedene Ansätze verwendet, namentlich der Gauß-Strahl und die Debye-Näherung, wie sie in [1] ausgeführt wird.

Der Gauß-Strahl ist eine einfache Möglichkeit, das Intensitätsfeld eines Lichtstrahl anzunähern. Dieser basiert auf der Annahme, dass die Lichtstrahlen nur schwach fokussiert sind. Beim 3D-Lasernanodruck, wo Objektive mit numerischen Aperturen von bis zu 1,4 zum Einsatz kommen, ist dies jedoch nicht der Fall. Daher wurde trotz dessen, dass deren Berechnung aufwändiger ist, die Debye-Näherung gewählt. Abbildung 6 zeigt die beiden normierten Intensitätsprofile im Vergleich. Wie deutlich die Simulationsergebnisse zum Teil voneinander abweichen, sieht man in Abbildung 7.

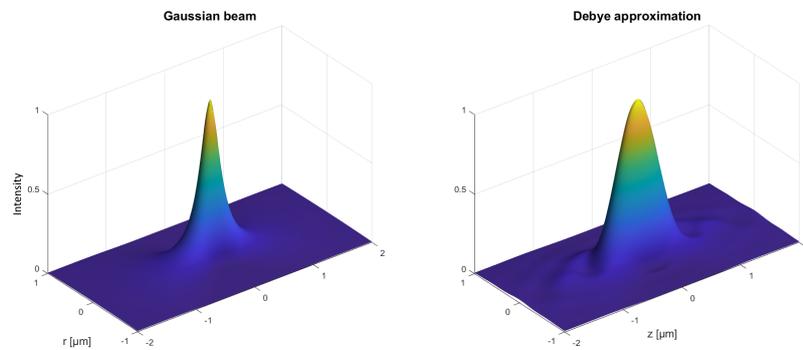


Abbildung 6: Vergleich der Intensitätsprofile des Gauß-Strahls und der Debye-Näherung; Eigene Abbildung

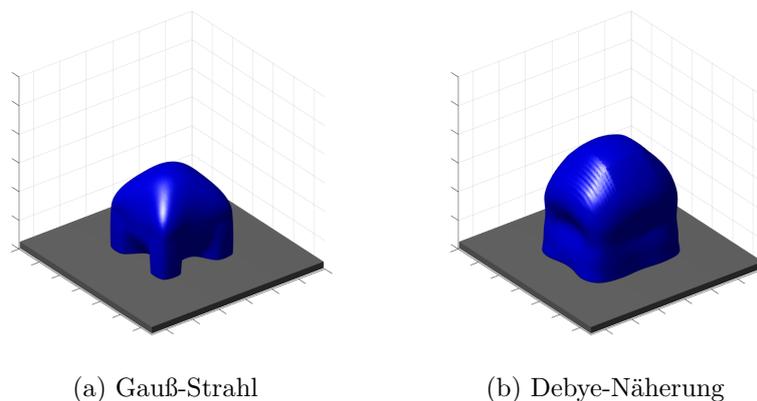


Abbildung 7: Vergleich der Simulation eines Tischdrucks bei 1PA mit dem Gauß-Strahl und der Debye-Näherung; Eigene Abbildung



### 3.2.2. Eingabe der 3D-Struktur

Das Einlesen der 3D-Strukturen geschah über das GWL-Dateiformat, das mit dem Programm DeScribe der Nanoscribe GmbH aus STL-Dateien generiert werden kann. Neben Metadaten enthält es Positionierungs- und Schreibanweisungen. In Tabelle 1 ist ein Auszug aus einer GWL-Datei abgebildet. In den ersten 3 Spalten sind die  $x$ -,  $y$ - und  $z$ -Koordinaten in Mikrometern angegeben. Die vierte Zeile gibt die Laserleistung in Prozent an.

```

% Slice 1/23, Z = 0
5.0   -4.3  0  0
4.5   -4.3  0  0
4.5   -4.3  0 100
2.625 -4.3  0 100
2.625 -4.3  0  0
2.125 -4.3  0  0
Write

5.0   -4.5  0  0
4.5   -4.5  0  0
4.5   -4.5  0 100
2.625 -4.5  0 100
2.625 -4.5  0  0
2.125 -4.5  0  0
Write

AddZDrivePosition 0.3
% Slice 2/23, Z = 0.3
4.7   -4.3  0  0
4.2   -4.3  0  0
4.2   -4.3  0 100
2.425 -4.3  0 100
2.425 -4.3  0  0
1.925 -4.3  0  0
Write

```

Tabelle 1: Auszug aus einer GWL-Datei für eine Struktur aus zwei parallele Linien in der untersten Ebene und einer höhergelegenen, an einem Ende überstehenden Linie

Die Zeilen, in denen die Leistung gleich null ist, sind Positionierungsanweisungen, die sicherstellen, dass die Galvo-Spiegel beschleunigen und den Schreibvorgang mit einer konstanten Geschwindigkeit vollführen können. Unter Einbezug einer bestehenden Bibliothek, die sich um die Auflösung der Makros kümmert, wird daraus ein MATLAB<sup>®</sup>-Array im Format von Tabelle 2 generiert, wobei jede Zeile einen Schreibvorgang enthält.

Hierbei ist zu beachten, dass die Werte in diesem Format nicht in Mikrometern angegeben sind, sondern Koordinaten innerhalb der Raummatrix angeben. Die ersten drei Spalten beschreiben die  $x$ -,  $y$ - und  $z$ -Koordinaten des Startpunkts. Da beim 3D-Druck immer nur horizontal geschrieben wird, reichen die zwei Spalten fünf und sechs aus, um den Endpunkt relativ zum Startpunkt angeben.

Das Verfahren zur Umwandlung der GWL-Werte in Koordinaten soll anhand des vorliegenden

Beispiels erklärt werden. Aus der GWL-Datei können diejenigen Zeilen extrahiert werden, bei denen die Leistung 100 % beträgt. Damit erhält man die Start- und Endpunkte aller Schreibvorgänge. Aus ihnen werden die kleinsten  $x$ -,  $y$ -, und  $z$ -Werte bestimmt. Im Beispiel sind dies  $x_{min} = 2,425 \mu\text{m}$ ,  $y_{min} = -4,5 \mu\text{m}$  und  $z_{min} = 0 \mu\text{m}$ . An diese Stelle wird der Ursprung der abbildenden Raummatrix gesetzt, um ausschließlich positive Bildkoordinaten zu erhalten. Mit der Formel (1), die entsprechend für  $y$  und  $z$  gilt, werden alle Angaben in Mikrometern zu Koordinaten der Raummatrix umgewandelt. In unserem Beispiel beträgt die Auflösung 25 nm.

$$x_{Bild} = \frac{x_{Original} - x_{min}}{Auflösung} \quad (1)$$

Als letzter Schritt werden für jeden Schreibvorgang die  $x$ -,  $y$ - und  $z$ -Koordinaten der Startpunkte in die Tabelle eingetragen. Für die Endpunkte werden je die Differenzen  $x_{Ende} - x_{Start}$  und  $y_{Ende} - y_{Start}$  eingetragen. Da beide Punkte immer in einer Ebene liegen, kann die Differenz der  $z$ -Koordinaten entfallen. Bei der Abbildung der Mikrometerwerte auf Matrixkoordinaten kann es je nach Auflösung zu Rundungsfehlern kommen, deren Einfluss in 5.1 untersucht wurde.

x	y	z	X <sub>off</sub>	Y <sub>off</sub>
83	8	0	-75	0
83	0	0	-75	0
71	8	12	-71	0

Tabelle 2: MATLAB<sup>®</sup>-Tabelle mit geparsten Daten aus der GWL-Datei

### 3.2.3. Akkumulation

Im Kern der Anwendung ist die Akkumulation mittels einer Schleife umgesetzt. Andere Verfahren, wie eine parallelisierte Variante oder der Rückgriff auf den Fourierraum wie in [1], sind für eine Animation ungeeignet, da hierbei nicht nur das Endergebnis, sondern auch aufeinander aufbauende Zwischenstände der Akkumulation von Interesse sind.

```

1  count = 0;
2  for line = lines
3      for pos = line_rastering(line(1:2), line(4:5))
4          if floor(mod_f(count, cam.step)) == 0
5              update_plot();
6              take_picture();
7          end
8          acc_space(pos(1)+1 : pos(1)+x_dim, ...
9                  pos(2)+1 : pos(2)+y_dim, ...
10                 platform_height+max(2,line(3)-z_dim_h+3) : ...
11                 line(3)+z_dim_h+1+platform_height)...
12         = D( : , : , max(1,z_dim_h-line(3)) : end) ...
13         + acc_space(pos(1)+1 : pos(1)+x_dim, ...
14                   pos(2)+1 : pos(2)+y_dim, ...
15                   platform_height+max(2,line(3)-z_dim_h+3) : ...
16                   line(3)+z_dim_h+1+platform_height);
17         count = count + 1;
18     end
19 end

```

Die Daten aus Tabelle 2 werden dem Programm in der Variable `lines` übergeben. Die einzelnen Felder, die in Abbildung 8 rot markiert sind, werden in Zeile 3 aus Start- und Endfeld jedes Schreibvorgangs berechnet. In Zeile 8 bis 16 wird die Dosismatrix `D` an den jeweils vorgegebenen Koordinaten zur Raummatrix `acc_space` hinzuaddiert. Dabei ist die Indizierung in der  $z$ -Dimension nötig, um den Kontakt der ersten gedruckten Schicht zur Substratplatte korrekt abzubilden.

### 3.2.4. Kameraeinstellung

Zu Beginn der Simulation übergibt man dem Programm die Dauer und Bildfrequenz der zu erstellenden Animation. Anhand der Daten aus Tabelle 2 kann die Anzahl der Akkumulationsschritte, das sind die Durchläufe der inneren Schleife, vorab berechnet werden.

$$\text{Akkumulationsschritte} = \sum_i (1 + \max\{x_{off}(i), y_{off}(i)\})$$

Die Laufvariable  $i$  indiziert dabei die Zeilen aus Tabelle 2. Aus der Anzahl der Akkumulationsschritte, der Bildfrequenz und der Dauer lässt sich berechnen, wie groß ein Kameraschritt ist, d. h. nach wie vielen Akkumulationsschritten je eine Aufnahme erzeugt wird. Dies erledigen die Zeilen 4 bis 7 im Programm.

Für einen besseren Überblick gibt es die Möglichkeit, während der Animation eine Kamerafahrt zu vollführen. Parametrisiert wird diese, indem man in einer Liste übergibt, in welchem Winkel die Kamera zu bestimmten Zeitpunkten auf die Struktur gerichtet sein soll. Dazwischen wird die Bewegung linear interpoliert.

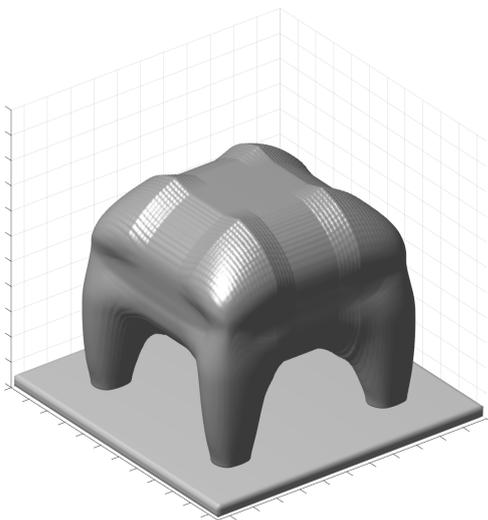
Zur Darstellung der Daten aus der Raummatrix waren wir an einem Weg interessiert, die Bereiche zu visualisieren, in denen die deponierte Lichtdosis die Schwellendosis überschreitet. Dabei hilft die Beobachtung, dass die Oberfläche der simulierten 3D-Struktur eine Isofläche ist. In einem Skalarfeld bezeichnet eine Isofläche eine Fläche, die aus allen Punkten besteht, die gleich einem festgelegten Wert sind. Zwar ist die Raummatrix, anders als ein Skalarfeld, nicht kontinuierlich, in MATLAB<sup>®</sup> kam aber die Möglichkeit gelegen, mittels der `isosurface`-Funktion eine Isofläche durch Interpolation anzunähern.

## 4. Ergebnisse

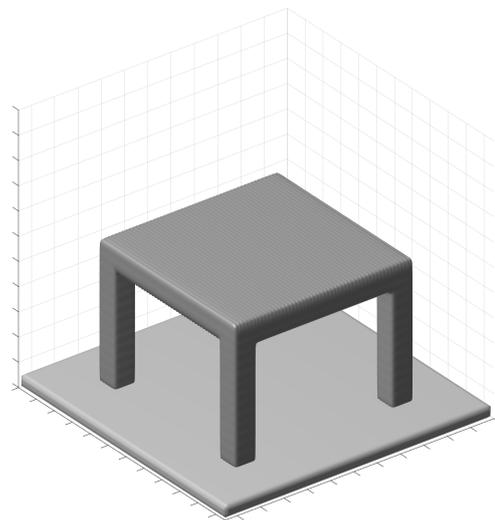
Mit dem bestehenden Simulationsprogramm untersuchten wir, wie die Präzision, die je mit der 1- und 2-Photonen-Absorption realisiert werden kann, durch Akkumulationseffekte begrenzt wird. Als Testobjekt diente ein kleiner Tisch. Im Ideal ist er  $15\ \mu\text{m}$  breit und  $11\ \mu\text{m}$  hoch. Die Beine und die Tischplatte haben jeweils eine Dicke von  $2\ \mu\text{m}$ . Als Referenz beträgt die Größe des Gitters im Hintergrund der Abbildungen ebenfalls  $2\ \mu\text{m}$ .

Die Animationsergebnisse wurden auf dem Symposium der Hector Fellow Academy präsentiert. Dessen Aufzeichnung lässt sich online anschauen [4]. Dort kann man die hier gezeigten Bilder auch in Bewegung und aus verschiedenen Kameraperspektiven sehen.

Im simulierten Druck kam ein Laser der Wellenlänge  $780\ \text{nm}$  zum Einsatz, der durch ein planachromatisches Objektiv mit 63-facher Vergrößerung und einer numerischen Apertur von 1,4 fokussiert wurde. Geschrieben wurde mit einer Geschwindigkeit von  $145\ \frac{\text{mm}}{\text{s}}$ .



(a) 1-Photon-Absorption



(b) 2-Photonen-Absorption

Abbildung 9: Vergleich der Simulation eines Tischdrucks bei 1PA und 2PA; Abbildung erstellt zusammen mit Richard Wagner

In der Simulation des Druckes mit 2PA beträgt die maximale Lichtdosis, die der Laser bei einem Schreibvorgang im Lack deponiert, 295 % der Schwellendosis. Bei der Simulation mit 1PA ist die Schwellendosis so weit erhöht, dass an den Stellen, die dem Laser am stärksten ausgesetzt sind, nach einmaliger Lichteinwirkung nur 41 % der Schwellendosis erreicht wird. Das führt dazu, dass der Laser, wenn er die Tischplatte das erste mal linienweise abfährt, nur in der Simulation mit 2PA feste Strukturen im Lack generiert. In der anderen Simulation kommt es erst beim zweiten Durchlauf zur Ausbildung der Tischplatte. Die Abbildungen 12 und 13 im Anhang zeigen die Zwischenstände der Simulationen zu genau diesen Zeitpunkten.

Die gewählte Schwellendosis für die Simulation mit 2PA orientierte sich an realen Werten. Bei der 1PA hingegen wurde die Schwellendosis so gewählt, dass das Ergebnis möglichst noch als Tisch zu erkennen ist, ohne diese zu hoch zu setzen, sodass die in Kapitel 2.4 erwähnte Ungenauigkeit durch Diffusion

zu groß wird. Es gab hier keine Werte zur Orientierung, da die 1PA beim 3D-Lasernanodruck in der Praxis nicht verwendet wird. Diese Simulation ist also keine Nachbildung eines realen Druckvorgangs, sondern eine allgemeine, theoretische Betrachtung.

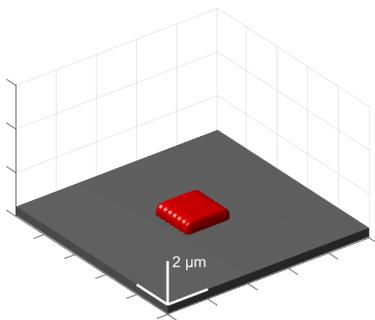
Die Auflösung wurde bei beiden Simulationen auf 100 nm gesetzt. Zusätzlich wurden die Dimensionen der Dosismatrix  $r_r$  und  $r_z$  aus Abbildung 8 reduziert, um die zeitaufwändige Berechnung der Animationen zu verkürzen.  $r_z$  betrug 15  $\mu\text{m}$ , das heißt, vom Fokuspunkt aus wurde der Einfluss des Lasers in axialer Richtung noch bis in diese Entfernung berechnet.  $r_r$  betrug dagegen nur 4  $\mu\text{m}$ . Dieses frühzeitige Abschneiden in transversaler Richtung führte zur Bildung der Kanten, die beim Blick auf 9a ins Auge fallen.

## 5. Diskussion

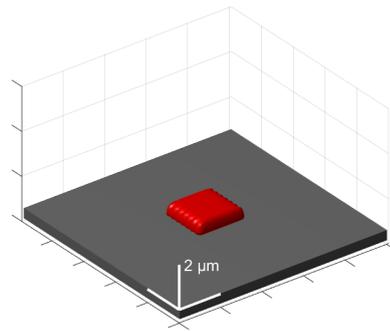
### 5.1. Auflösung

Zwecks kürzerer Rechenzeiten wurde die Auflösung der Simulation auf 100 nm gesetzt, was nur etwas kleiner ist als die Größenordnung der Voxel. Um daher zu untersuchen, wie sehr die Auflösung möglicherweise Rundungsfehler bewirkt, wurden vier Testsimulationen mit den Auflösungen 25 nm und 100 nm, sowie einem Hatching von 300 nm und 333 nm ausgeführt. Die Ergebnisse sind in Abbildung 10 aufgelistet. In 10a und 10b ist zu sehen, dass die Konturen der Voxel, die durch das Hatching entstehen, unabhängig von der Auflösung gleichermaßen ausgebildet sind. Auch eine Auflösung von 100 nm ist nicht zu klein, um dies beobachten zu können.

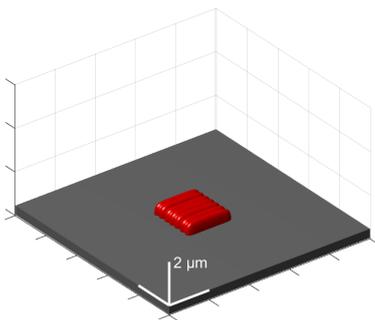
Dagegen treten bei falscher Wahl der Hatching- und Slicingwerte deutliche Unterschiede durch Rundungsfehler auf. Sind Hatching und Slicing kein Vielfaches der Auflösung und ist die Auflösung zusätzlich zu groß gewählt, kommt es zu periodischen Verschiebungen wie in 10d. Bei einer Auflösung von 25 nm sind die Rundungsfehler zwar auch zu erkennen, aber schwächer ausgeprägt. In der Simulation des Tisches (Abbildung 9) ist der angesprochene Effekt jedoch kein Problem, da Hatching mit 200 nm und Slicing mit 300 nm ein Vielfaches der verwendeten Auflösung sind.



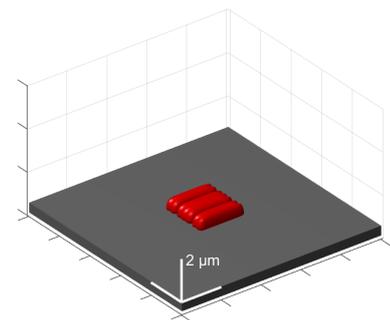
(a) Auflösung 25 nm, Hatching 300 nm



(b) Auflösung 100 nm, Hatching 300 nm



(c) Auflösung 25 nm, Hatching 333 nm



(d) Auflösung 100 nm, Hatching 333 nm

Abbildung 10: Einfluss der Auflösung auf die Simulationsergebnisse bei verschiedenen Hatchings; Eigene Abbildung

## 5.2. Laufzeit

Ein einschränkender Faktor, der u. a. zu den Kanten in Abbildung 9a führte, war die Laufzeit, die das Erstellen von Animationen jener Größe brauchte. So konnten wir bisher nicht versuchen, die radiale Begrenzung  $r_r$  des berechneten Feldes bei der Simulation auf einen größeren Wert als  $4\mu\text{m}$  zu setzen. Eine Neuberechnung des Fokus in einem größeren Bereich wäre zu aufwändig gewesen.

Es gab die Idee, die Ausführungszeit der Schleife, in der die Akkumulation stattfindet, zu verkürzen, indem die Summierung der Dosis zwischen der Aufnahme von zwei Bildern parallelisiert wird. Das Optimierungspotenzial durch diesen Ansatz ist jedoch aus zwei Gründen begrenzt. Zum einen ergab ein kurzer Test, dass die Akkumulation einen ähnlichen Rechenaufwand benötigt wie die Grafikberechnungen zur Darstellung der Isofläche. Als zweites sind die Akkumulationsschritte zwischen zwei Bildern zu wenige, sodass sich eine Parallelisierung nicht lohnt. Die sequenzielle Variante hat den Vorteil, dass innerhalb eines Akkumulationsschritts nur Berechnungen in einem Bereich der Größe der Dosismatrix ausgeführt werden müssen. Diese ist kleiner als die vollständige Raummatrix. In der parallelen Variante könnte man zwar einige Schritte gleichzeitig ablaufen lassen, dies müsste aber in unterschiedlichen Raummatrizen geschehen. Durch das Zusammenführen geht der Vorteil bei zu wenigen Akkumulationsschritten wieder verloren. Die Animation des Tischdrucks mit einer Bildfrequenz von 30 Hz und einer Dauer von 30 s zum Beispiel hatte zwischen zwei Bildern nur 190 Akkumulationsschritte.

## 6. Ausblick

Mit diesem Projekt konnten wir zeigen, dass die feinen und gleichzeitig komplexen Strukturen, die man mit dem 3D-Lasernanodruck z. B. für optische Anwendungen drucken will, mit der 1-Photon-Absorption nicht realisiert werden können.

Die 2-Photonen-Absorption macht dies zwar möglich, zieht aber andere Nachteile nach sich. Da die Wahrscheinlichkeit sinkt, dass ein Initiator-molekül angeregt wird, werden zum Ausgleich höhere Lichtintensitäten benötigt, die nur durch Femtosekunden-Pulslaser erreicht werden können. Da diese sehr teuer sind, ist eine Herstellung von 3D-Lasernanodruckern in großer Serie bisher nicht denkbar.

Aus diesem Grund wird momentan am Institut für Angewandte Physik des KIT an der 2-Stufen-Absorption geforscht. Hierbei werden Initiator-moleküle verwendet, die zwischen angeregtem und nicht angeregtem Zustand ein weiteres Energieniveau aufweisen. Trifft ein Photon auf einen solchen Initiator, kann er für eine bestimmte Zeit in diesem Zwischenzustand bleiben, bevor er in den Grundzustand zurückfällt. Ein zweites Photon hat somit mehr Zeit, den Zerfall des Initiators anzustoßen. Dadurch bleibt der so wichtige Nichtlinearitätswert von 2 erhalten, während die Wahrscheinlichkeit der Initiierung bei gleicher Lichtintensität steigt.

## A. Anhang

### Inhaltsverzeichnis

A.1. Auszüge aus der Animation . . . . .	I
A.2. Programmcode . . . . .	II
A.2.1. Quelltext des Scripts . . . . .	II
A.2.2. Funktionen . . . . .	IV
A.3. Danksagung . . . . .	VII
A.4. Literatur . . . . .	VIII
A.5. Selbstständigkeitserklärung . . . . .	IX

### A.1. Auszüge aus der Animation

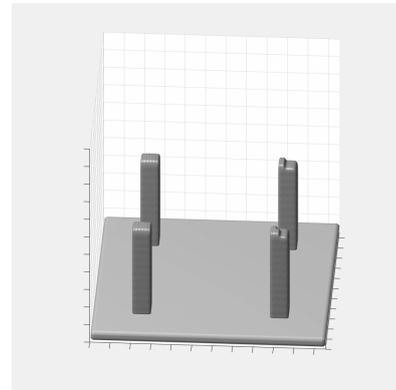
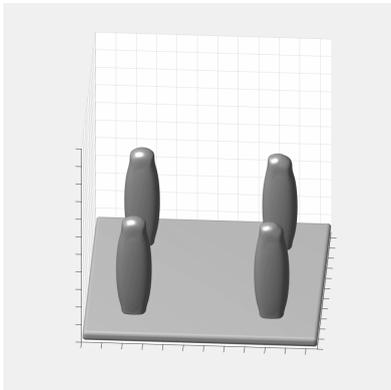


Abbildung 11: Aufnahme der Simulation zum Zeitpunkt des Drucks der Tischbeine

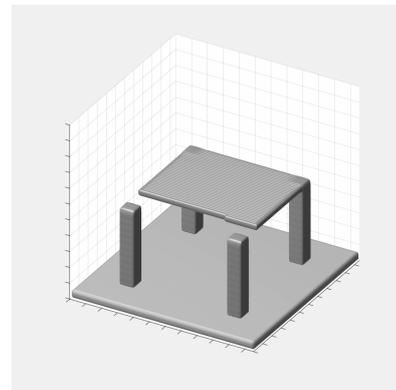
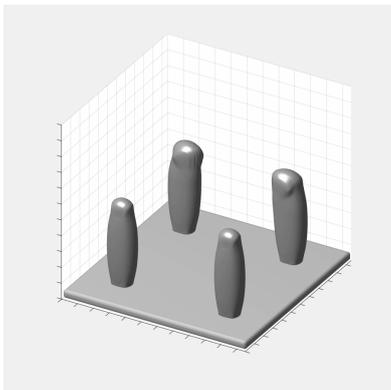


Abbildung 12: Aufnahme der Simulation zum Zeitpunkt des Drucks der ersten Schicht der Tischplatte

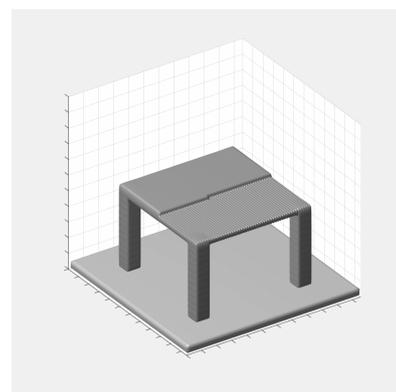
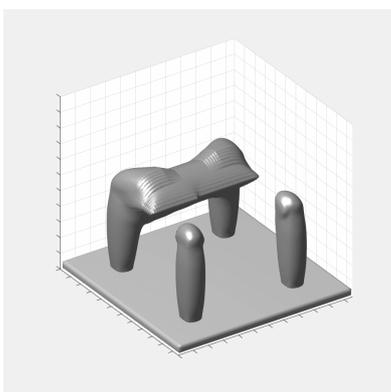


Abbildung 13: Aufnahme der Simulation zum Zeitpunkt des Drucks der zweiten Schicht der Tischplatte

## A.2. Programmcode

### A.2.1. Quelltext des Scripts

```

1  %% Animation of accumulated dose in the threshold model
2  %{
3      1. Set input parameters
4          1.1. Program parameters
5          1.2. Focus parameters
6          1.3. Laser and photoresist parameters
7          1.4. Camera parameters
8      2. Select focus
9      3. Select gwl job
10     4. Select camera trajectory
11     5. Animation
12  %}
13
14  %% 1. Set input parameters
15  %    1.1. Program parameters
16
17  pro.res = 100e-9;           % Resolution    [m]
18  pro.r_r = 4e-6;           % Radial Range [m]
19  pro.r_z = 20e-6;         % Axial Range  [m]
20
21  %    1.2. Focus parameters
22
23  foc.lam = 780e-9;         % Vacuum Wavelength [m]
24  foc.n = 1.52;            % Diffraction Index of Medium [ ]
25  foc.NA = 1.4;           % Numerical Aperture [ ]
26  foc.tubus = 165e-3;      % Tubus Length [m]
27  foc.M = 63;             % Magnification of Objective [ ]
28  foc.t = 0.7;           % Transmission Coefficient [ ]
29  foc.w = foc.NA*foc.tubus/foc.M; % Laser Radius on Objective [m]
30  foc.f = foc.tubus/foc.M; % Focal Length [m]
31
32  %    1.3. Laser and photoresist parameters
33
34  l.nonlin = 2;           % Nonlin. factor of photoresist [ ]
35
36  %    1.4. Camera parameters
37
38  cam.thresh = 1;
39  cam.color = 'red';
40  cam.tick_res = 2e-6;    % Distance between ticks on the axis [m]
41  cam.fps = 30;          % Frame rate of the animation [1/s]
42  cam.duration = 10;     % Duration of the animation [s]
43  cam.filename = "filename"; % Save animation to "filename.gif"
44
45  %% 2. Select focus
46
47  disp('Select calculated focus...')
48  [file,path] = uigetfile('.mat','Select focus','./foci/');
49  if isequal(file,0)
50      disp('User selected Cancel');
51  else
52      disp(['User selected ', fullfile(path,file)]);
53  end
54  load(fullfile(path,file),'E');
55

```

```

56 D = E .^ 1.nonlin;
57 D = D ./ max(D(:));
58
59 %% 3. Select gwl job
60
61 addpath(genpath('matlab-gwlParser')); % Make sure that the library folder is included in path
62 % Import gwl file
63 disp('Select gwl job (raw or parsed, try the different file types!)...')
64 [file,path,indx] = uigetfile({'*.gwl';'*parsed.mat'},'Select gwl job','./gwl_jobs/');
65 if isequal(file,0)
66     disp('User selected Cancel');
67 else
68     disp(['User selected ', fullfile(path,file)]);
69 end
70 if indx == 1
71     gwl = gwlParserFunctionFP(file,path);
72 elseif indx == 2
73     load(fullfile(path,file), 'gwl');
74 else
75     disp('Make sure to select the correct file types!');
76 end
77 % Convert gwl data
78 %{
79     .gwl file:
80         1           2 3 4 5           6 7
81         commands  x  y  z  power  ?  ?
82     convert to:
83         1 2 3 4           5           6
84         x y z x_off y_off z
85 %}
86 scribe = gwl(gwl(:,1) == 0,:); % Cut lines containing commands
87 scribe = scribe(scribe(:,5) ~= 0 | circshift(scribe(:,5) ~= 0, 1),:); % Get rid of positioning
88     ↪ movements
89 scribe = scribe(:,2:4); % Retain fields x,y,z only
90
91 scribe = scribe*1e-6;
92 scribe(:,3) = scribe(:,3)*1e3; % Convert z to microns
93
94 scribe = round((scribe - min(scribe))/pro.res);
95
96 pro.x_max = max(scribe(:,1));
97 pro.y_max = max(scribe(:,2));
98 pro.z_max = max(scribe(:,3));
99
100 scribe = reshape(scribe',6,[]);
101 scribe([4 5],:) = scribe([4 5],:) - scribe([1 2],:);
102
103 %% 4. Select camera trajectory
104
105 % Import trajectory file
106 disp('Select camera trajectory...')
107 [file,path] = uigetfile('*.*mat','Select camera trajectory','./trajectories/');
108 if isequal(file,0)
109     disp('User selected Cancel');
110 else
111     disp(['User selected ', fullfile(path,file)]);
112 end
113 load(fullfile(path,file),'traj');

```

```

113 % Calculate camera angles
114 addpath(genpath('interpolation'));
115 accumulation_steps = sum( max(abs(scribe(4,:)),abs(scribe(5,:))) + 1 ); % Calculate the sum of the
↳ steps line_rastering() will generate for each written line
116 cam.fps = min(cam.fps, accumulation_steps/cam.duration); % Frame rate (together with length) may
↳ demand more frames than there are steps to make
117 animation_steps = cam.duration * cam.fps;
118 cam.step = accumulation_steps/animation_steps;
119
120 cam.angles = zeros(animation_steps+1,2); % +1 because there will be one additional last frame
121 for i = 2:size(traj,1)
122     xx = round(traj([i-1 i],1)*animation_steps)+1; % Get time now and now-1
123
124     aa = traj([i-1 i],2);
125     az = linear_interpolation(xx,aa); % Azimuth change
126
127     ee = traj([i-1 i],3);
128     el = linear_interpolation(xx,ee); % Elevation change
129
130     cam.angles(xx(1):xx(2)-1,:) = [az(1:end-1); el(1:end-1)]';
131 end
132 cam.angles(end,:) = traj(end,2:3); % Last frame
133
134 %% 5. Animation
135
136 animation(pro, D, scribe, cam);

```

## A.2.2. Funktionen

```

1 function animation(pro, D, scribe, cam)
2     %% Allocate accumulation space
3     % Calculate voxel dimensions
4     x_dim = floor(2*pro.r_r/pro.res)+1;
5     y_dim = x_dim;
6     z_dim_h = floor(pro.r_z/pro.res)+1; % Note: only half
7     % Calculate accumulation space dimensions
8     x_max = pro.x_max;
9     y_max = pro.y_max;
10    z_max = pro.z_max;
11    x_top = x_max+x_dim;
12    y_top = y_max+y_dim;
13    z_top = z_max+z_dim_h+1; % +1 to fill the isosurface hole at the bottom
14    % Create substrate in space
15    platform_height = ceil(z_top/50);
16    platform = zeros(x_top, y_top, z_top + platform_height);
17    platform(:, :, 1:ceil(z_top/50)) = ones(x_top, y_top, ceil(z_top/50));
18    %Create accumulation space
19    acc_space = zeros(x_top, y_top, z_top+platform_height);
20
21    %% Accumulate while animation is performed
22    % Set up the plot
23    tic
24    fig = figure('Name','animation','NumberTitle','off', 'Position', [0 0 900 900]);
25    plot_x_ave = (1:x_top) * pro.res;
26    plot_y_ave = (1:y_top) * pro.res;
27    plot_z_ave = (1:z_top + platform_height) * pro.res;
28    axis equal
29    axis([0 y_top*pro.res 0 x_top*pro.res 0 z_top*pro.res])

```

```

30     xticks(0:cam.tick_res:y_top*pro.res);
31     yticks(0:cam.tick_res:x_top*pro.res);
32     zticks(0:cam.tick_res:(z_top + platform_height)*pro.res);
33     xticklabels({});
34     yticklabels({});
35     zticklabels({});
36     grid on
37     % Accumulate
38     number_of_pictures = 0;
39     count = 0;
40     for line = lines
41         for pos = line_rastering(line(1:2), line(4:5))
42             if floor(mod_f(count, cam.step)) == 0
43                 update_plot();
44                 take_picture();
45             end
46             acc_space(pos(1)+1 : pos(1)+x_dim, ...
47                     pos(2)+1 : pos(2)+y_dim, ...
48                     platform_height+max(2,line(3)-z_dim_h+3) : ...
49                     line(3)+z_dim_h+1+platform_height)...
50             = D( : , : , max(1,z_dim_h-line(3)) : end) ...
51             + acc_space(pos(1)+1 : pos(1)+x_dim, ...
52                       pos(2)+1 : pos(2)+y_dim, ...
53                       platform_height+max(2,line(3)-z_dim_h+3) : ...
54                       line(3)+z_dim_h+1+platform_height);
55             count = count + 1;
56         end
57     end
58     while number_of_pictures < size(cam.angles,1)-1
59         update_plot();
60         take_picture();
61     end
62     update_plot();
63     take_picture('last');
64     disp(['Animation took ' num2str(toc) ' seconds; Number of frames: '
65         ↪ num2str(number_of_pictures)]);
66
67     %% Private functions
68     function update_plot()
69         cla
70         % Plot printed object
71         p = patch(isosurface(plot_y_axe, plot_x_axe, plot_z_axe, acc_space, cam.thresh));
72         set(p, 'FaceColor', cam.color, 'EdgeColor', 'none');
73         % Plot platform
74         p2 = [patch(isosurface(plot_y_axe, plot_x_axe, plot_z_axe, platform, 0.5)), ...
75             patch(isocaps(plot_y_axe, plot_x_axe, plot_z_axe, platform, 0.5))];
76         set(p2, 'FaceColor', '#777777', 'EdgeColor', 'none');
77         % Set up camera
78         view(cam.angles(number_of_pictures+1,:))
79         camlight left
80         lighting gouraud
81     end
82
83     function take_picture(shot)
84         gif_filename = string(append(cam.filename, '.gif'));
85         frame = getframe(fig);
86         im = frame2im(frame);
87         [imind,cm] = rgb2ind(im,256);

```

```

87         if number_of_pictures == 0
88             imwrite(imind,cm, gif_filename , 'gif','Loopcount',1, 'DelayTime', 0.5);
89             elseif ~exist('shot','var')
90                 imwrite(imind,cm, gif_filename, 'gif','WriteMode','append', 'DelayTime', 1/cam.fps);
91                 elseif strcmp(shot,'last')
92                     imwrite(imind,cm, gif_filename, 'gif','WriteMode','append', 'DelayTime',
93                             ↵ 0.5);
94                     saveas(fig, append(cam.filename, '.png'), 'png');
95                 end
96             number_of_pictures = number_of_pictures + 1;
97         end
98     end
99     function [b] = mod_f(a,m)
100     % "Modulo" operation with decimal divisor
101         b = a - m.*floor(a./m);
102     end
103
104     function [xy] = line_rastering(start,delta)
105     % start: 2D-vector of two integers
106     % delta: = end_of_line - start_of_line; 2D-vector of two ints
107     x0 = start(1);
108     dx = delta(1);
109     y0 = start(2);
110     dy = delta(2);
111     if dy == 0
112         [x,y] = horizontal_d(x0,dx,y0);
113     elseif dx == 0
114         [y,x] = horizontal_d(y0,dy,x0);
115     elseif abs(dx) >= abs(dy)
116         [x,y] = diagonal_d(x0,dx,y0,dy);
117     else
118         [y,x] = diagonal_d(y0,dy,x0,dx);
119     end
120     xy = [x; y];
121 end
122
123 function [x,y] = horizontal_d(x0,dx,y0)
124     if dx >= 0
125         x = x0:x0+dx;
126     else
127         x = x0:-1:x0+dx;
128     end
129     y = zeros(1, abs(dx)+1) + y0;
130 end
131
132 function [x,y] = diagonal_d(x0,dx,y0,dy) % Assert: |x1-x0| >= |y1-y0|
133     x = horizontal_d(x0,dx,y0);
134     m = dy / abs(dx);
135     y = round(y0:m:y0+dy);
136 end

```

### **A.3. Danksagung**

Ganz herzlich wollen wir uns bei Tobias Messer und Pascal Kiefer bedanken, die uns während des Kooperationsprojekts intensiv betreut haben und bei Fragen stets zur Seite standen.

Für die Arbeit zur Modellierung der Laserstrahlen, auf die wir zurückgreifen durften, danken wir Richard Wagner, ebenso für seine Hilfe am visuellen Schliff der finalen Animationen.

Vielen Dank an Prof. Dr. Martin Wegener, der dieses Kooperationsprojekt ermöglicht und unsere Animationsergebnisse auf dem Symposium der Hector Fellow Academy präsentiert hat.

Des Weiteren möchten wir uns natürlich bei allen Kursleiterinnen und Kursleitern bedanken, die uns sechs Jahre lang begleitet und das Hector-Seminar mit ihrem Einsatz zu jener spannenden Zeit gemacht haben, auf die wir gerne zurückschauen.

Zum Schluss gilt unser Dank dem Ehepaar Dr. Hans-Werner und Josephine Hector, ohne deren Förderung durch die Hector-Stiftung die zahlreichen Einblicke in das wissenschaftliche Arbeiten, die wir im Laufe des Hector-Seminars erhielten, nicht möglich gewesen wären.

**A.4. Literatur**

- [1] Wagner, Richard: Bachelorarbeit *On the Simulation of Dose Accumulation in 3D Laser Nanoprinting*, 2022, KIT
- [2] Müller, Dr. Jonathan B.: Dissertation *Exploring the Mechanisms of Three-Dimensional Direct Laser Writing by Multi-Photon Polymerization*, 2015, KIT
- [3] Frenzel, Dr. Tobias: Dissertation *On 3D Chiral Mechanical Metamaterials*, 2020, KIT
- [4] Symposium der Hector-Fellow-Academy: *3D-Druck - Maßgeschneidert und auf Knopfdruck*, 2022, Referent Dr. Martin Wegener, Aufzeichnung: [https://www.youtube.com/watch?v=\\_dmUv47lMwk](https://www.youtube.com/watch?v=_dmUv47lMwk)
- [5] Gietz, Paul; Schierle, Prof. Dr. Werner; Stein-Bastuck, Dr. Reiner; Sternberg, Michael: *Elemente Chemie*, Schulbuch für die Kursstufe Baden-Württemberg, 2010, Ernst Klett Verlag

**A.5. Selbstständigkeitserklärung**

Hiermit versichern wir, dass diese Arbeit unter Beratung von Tobias Messer, Pascal Kiefer und Richard Wagner sowie Norbert Krieg selbständig verfasst wurde, keine anderen als die angegebenen Quellen und Hilfsmittel verwendet und Zitate kenntlich gemacht wurden.

---

Elias Gestrich

---

Dan Vierling