Adaptives Slicing

(Interiour/Exteriour/Microlayering)

Rebecca Heinsohn

10.06.2015

Inhaltsverzeichnis

Quelle

Titel:

ADAPTIVE HIGH-PRECISION EXTERIOR, HIGH-SPEED INTERIOR, LAYERED MANUFACTURING

Autor:

Emmanuel Sabourin

Institut:

Faculty of the Virginia Polytechnic Institute and State University

Zeit und Ort der Veröffentlichung:

Februar 1996 in Blacksburg, Virginia

Problemstellung

- der Fokus wird auf das FDM-Verfahren gelegt
- eine feine, genaue Oberfläche erfordert eine geringe Schichtdicke
- nimmt man eine große Schichtdicke, ist es zwar schnell, aber qualitativ schlechter

=> Ziel des Verfahrens:

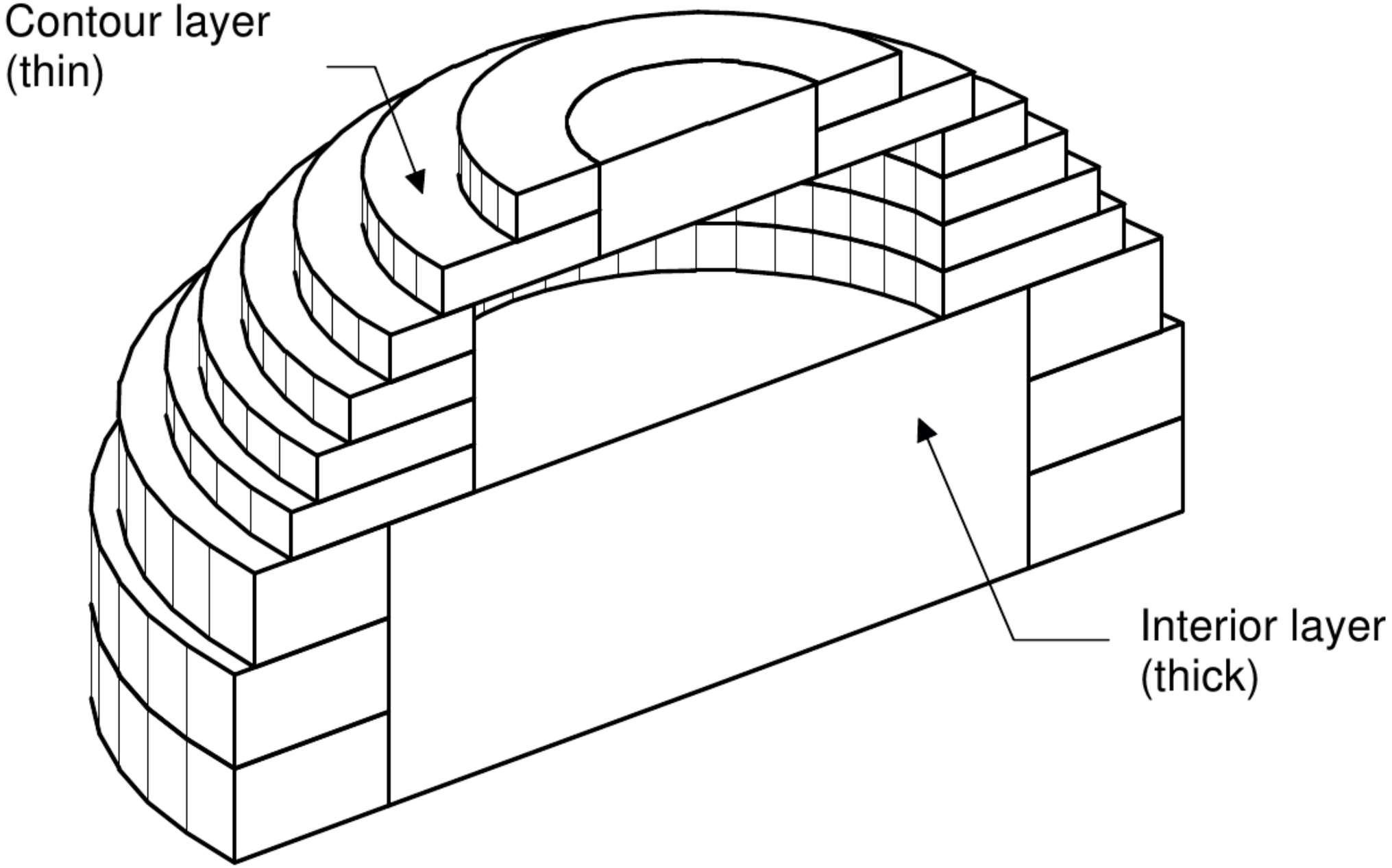
Weniger Druckzeit trotz guter Qualität

Definition/Idee

- Aufteilung des Objekts in Innen-/Außenbereich (Interiour/Exteriour)
- Außenbereich: dünne Schichten, die in ihrer Schichtdicke variieren, schmale Materialbahnen
- Innenbereich: dicke Schichten, breite Materialbahnen, Freilassen (Gitter)
  -> höhere Druckgeschwindigkeit, geringerer Materialverbrauch

=> schnellerer Druck, trotz hoher Präzision

Definition/Idee



Verfahren (Überblick)

- das Objekt wird in Slices mit größtmöglicher Schichtdicke unterteilt
- die Außenkonturen werden dann um einen gewissen Wert nach innen versetzt
- Außenbereich: adaptives Slicen in dünnere Slices: Schichtdicke verhältnismäßig zu Konturlinien

Verfahren (Ablauf)

Erster Teil:

1) Modell laden (.STL-Datei)
2) *flat areas* finden
3) Generierung der *thick layers*

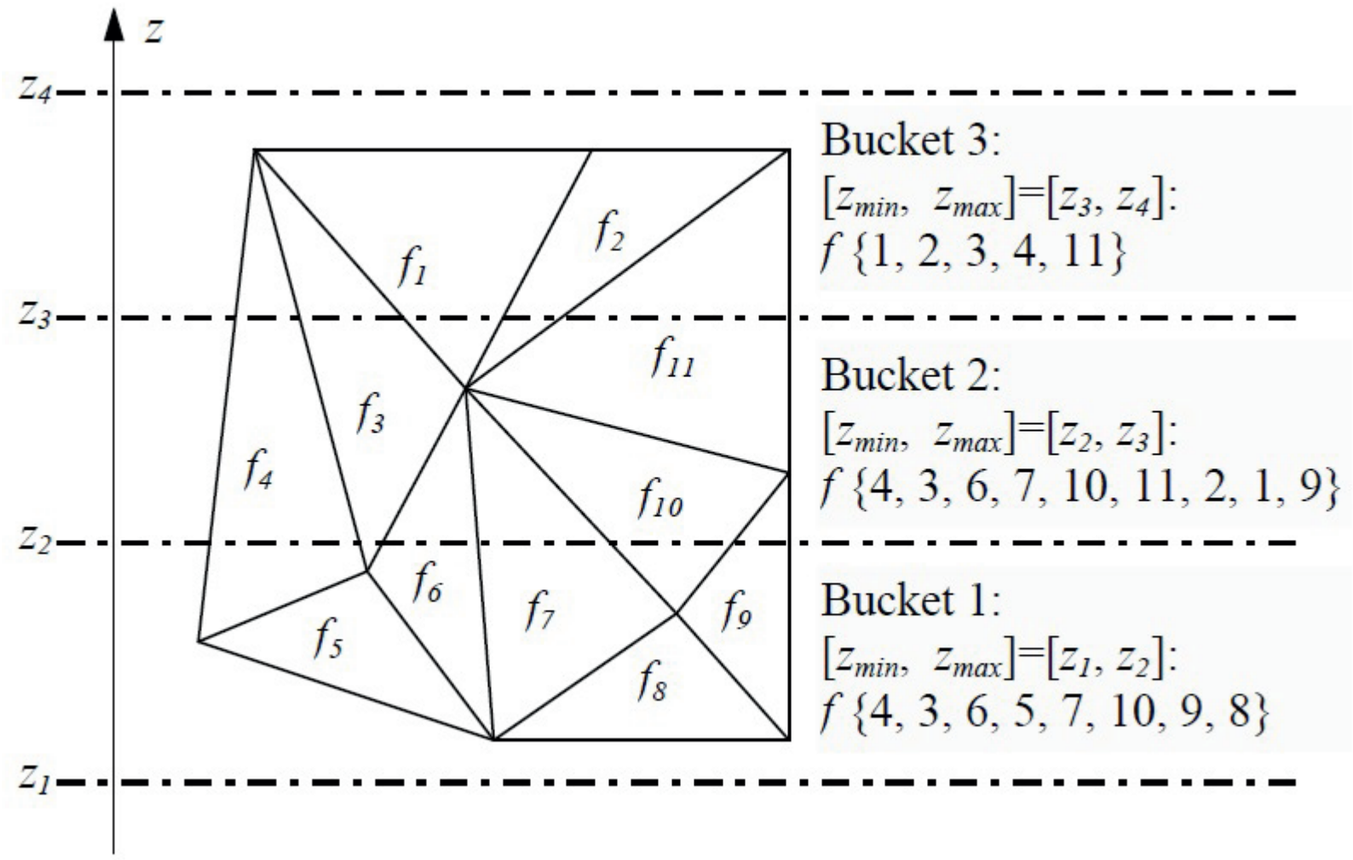
Zweiter Teil:

1) definieren von Innen- und Außenbereich für alle *thick layers*
2) Adaptives Slicing in den Außenbereichen

Verfahren (Ablauf)

Erster Teil: 1) Modell laden

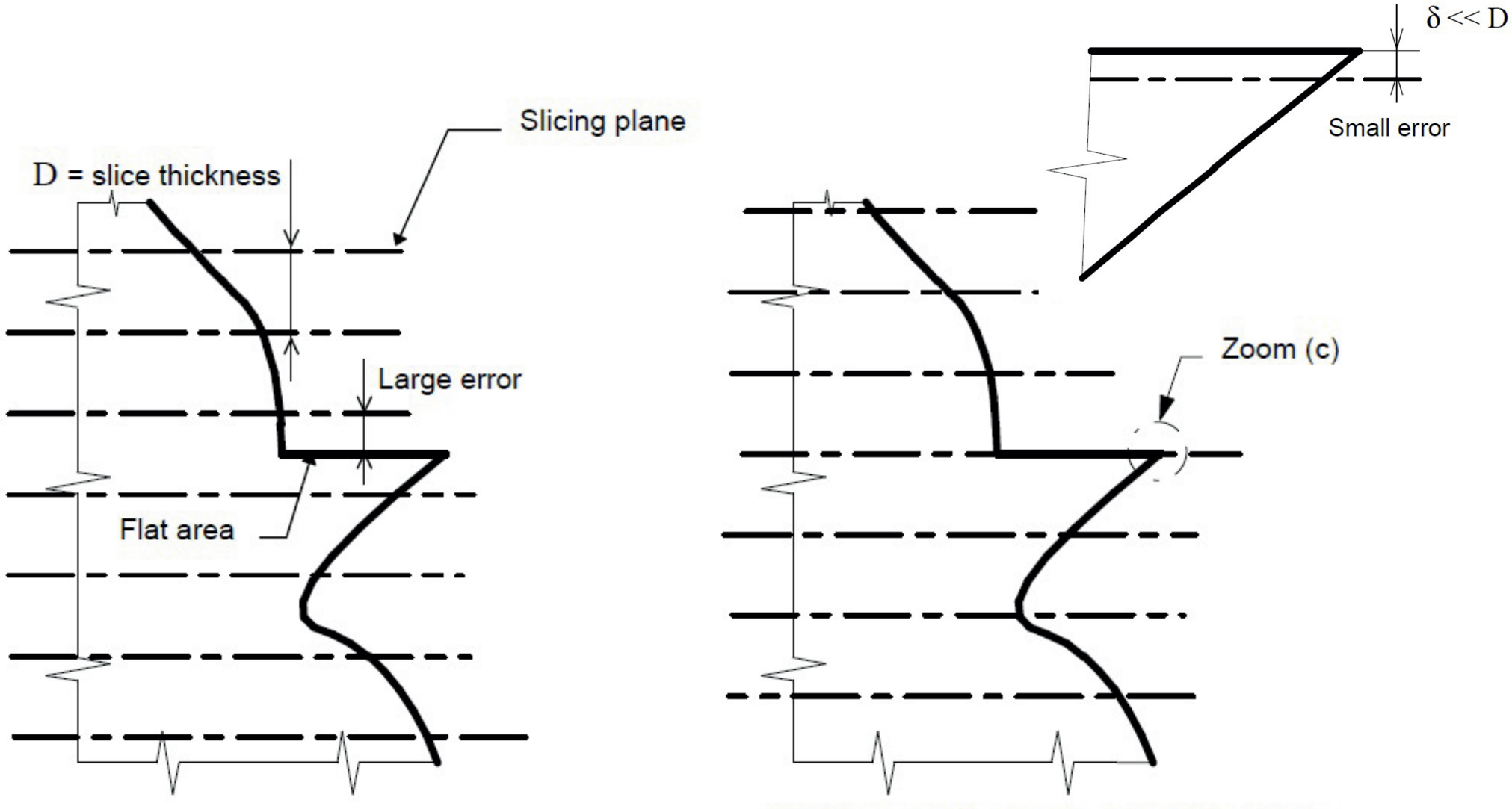
- *topology*: jedes *facet* enthält Referenzen zu seinen drei Scheitelpunkten und seinen drei Nachbar-*facets*
- *buckets*: sobald die *Slices* erstellt wurden, werden die *facets* in *buckets* sortiert



Verfahren (Ablauf)

Erster Teil: 2) *flat areas* finden

- *flat areas*: eine Stelle, an der mehrere *facets* eine horizontale Fläche bilden



Verfahren (Ablauf)

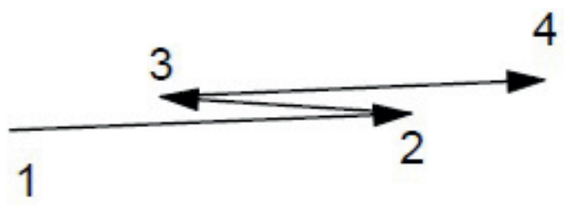
Erster Teil: 3) Generierung der *thick layers*

- Slicing-Höhen werden unter Berücksichtigung der max. Schichtdicke und der *flat areas* bestimmt
- Aufbau in Relation zur unteren Schicht:
  1) Slice-Höhe = untere Slice-Höhe+Standardhöhe
  2) Slice-Höhe = *flat area*-Höhe
- mit dem *marching algorithmus* werden alle möglichen Konturen einer Schicht berechnet

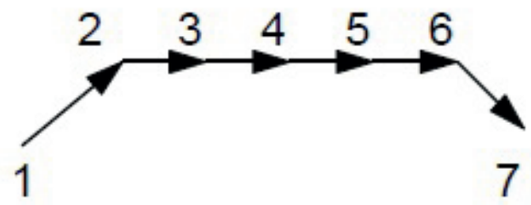
Verfahren (Ablauf)

Zweiter Teil: 1) Innen- und Außenbereich

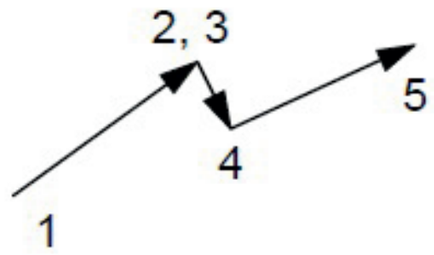
- Außenkontur optimieren



(a) Backward jerks



(b) Collinear edges



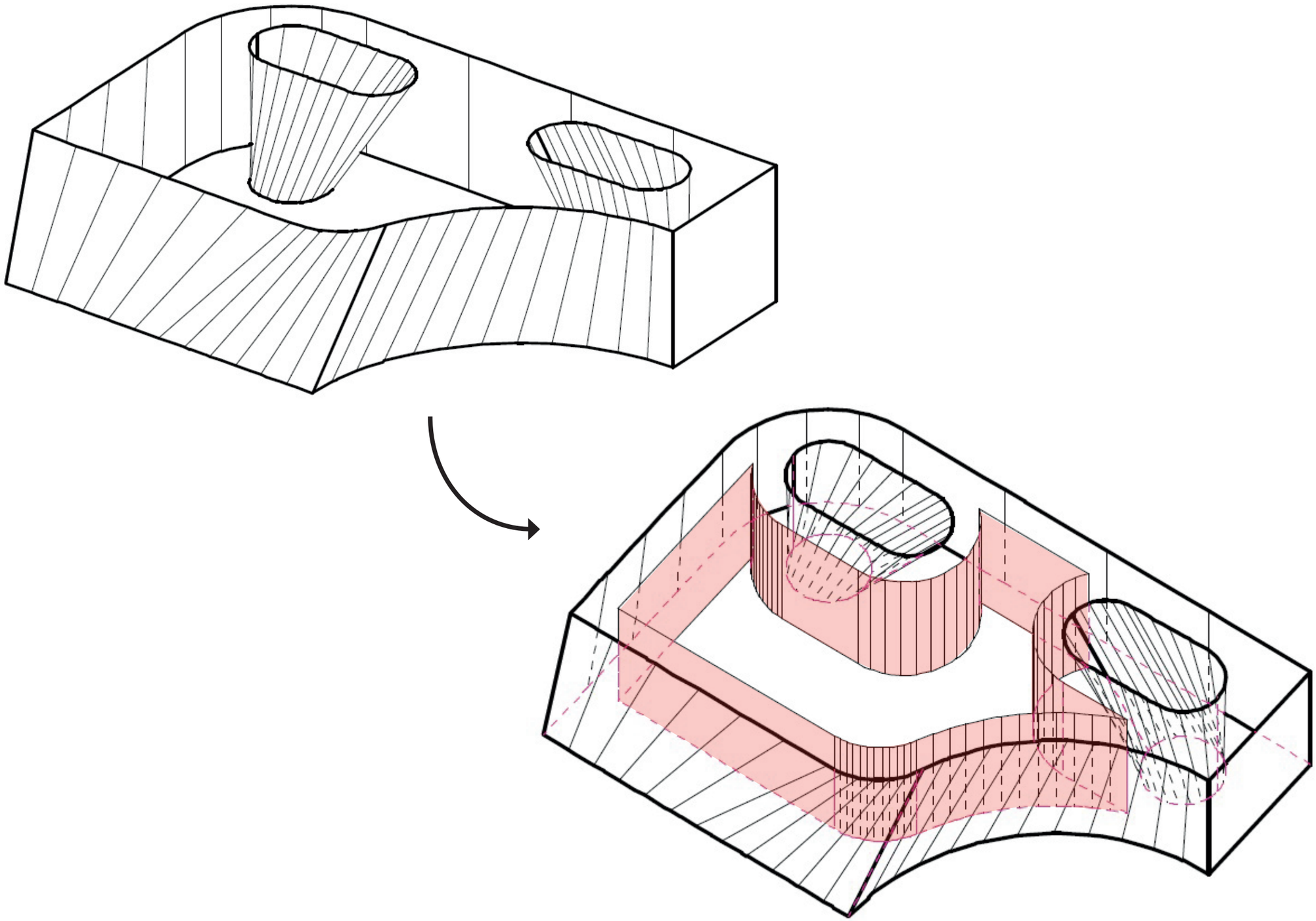
(c) Co-incident vertices

Verfahren (Ablauf)

Zweiter Teil: 1) Innen- und Außenbereich

- beginnt mit dem Verschieben der obersten Kontur der jeweiligen Schicht und wird auch bei allen weiteren durchgeführt
- ca. 3 mm Versatz nach innen
- neue Kontur wird senkrecht nach unten “durchgezogen”

Verfahren (Ablauf)



Verfahren (Ablauf)

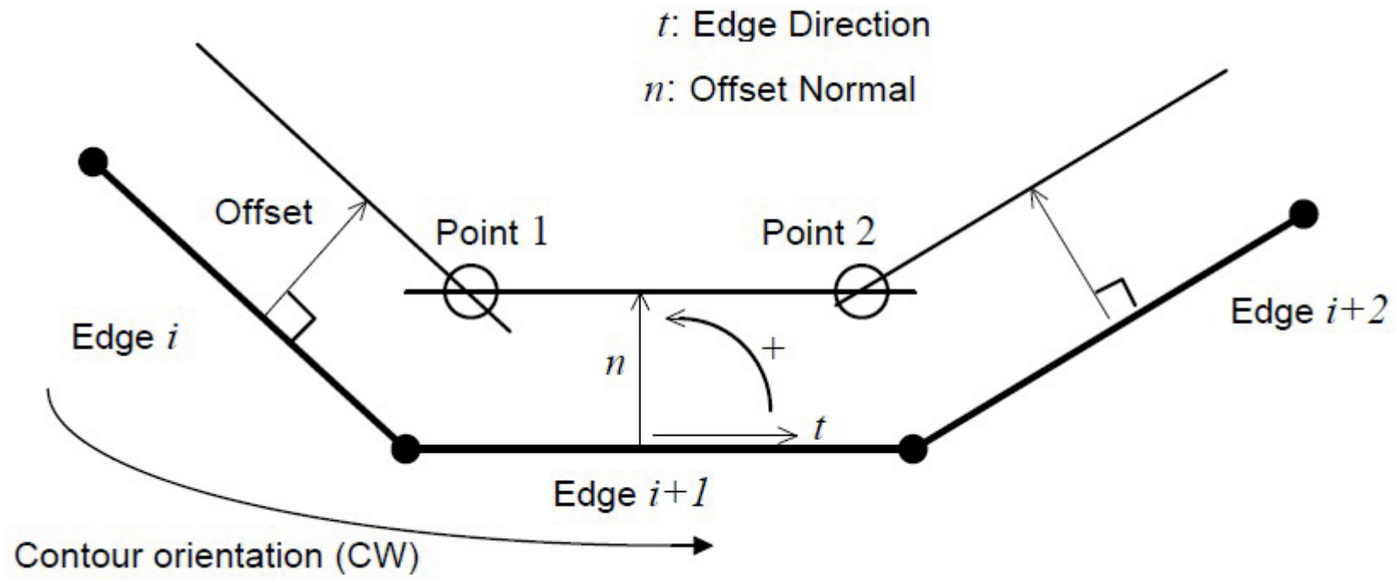
Zweiter Teil: 1) Innen- und Außenbereich

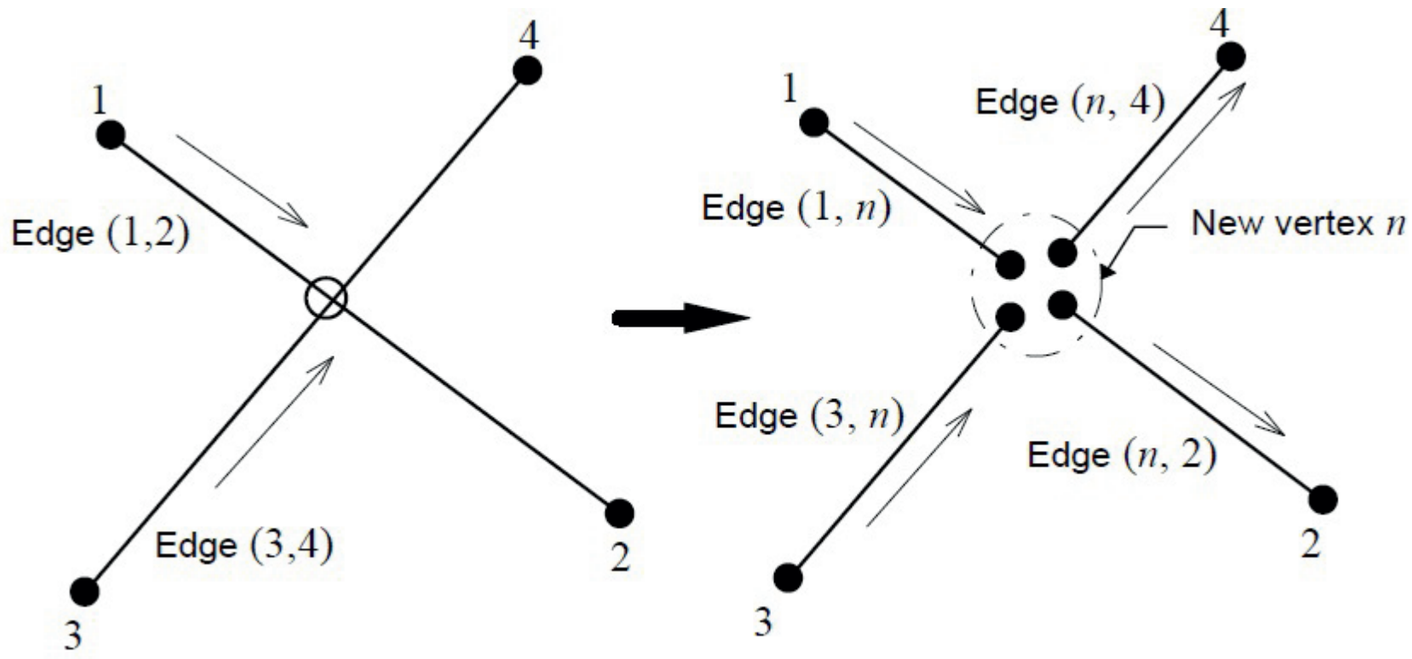
- Problem: durch Versatz nach innen könnten sich Konturen überschneiden, wo sie es vorher nicht getan haben

- Lösung:
  1) alle Überschneidungen identifizieren (“*contour marching algorithm*”)
  2) dort wo sich Kanten überschneiden, einen neuen Knoten hinzufügen

Verfahren (Ablauf)

Zweiter Teil: 1) Innen- und Außenbereich





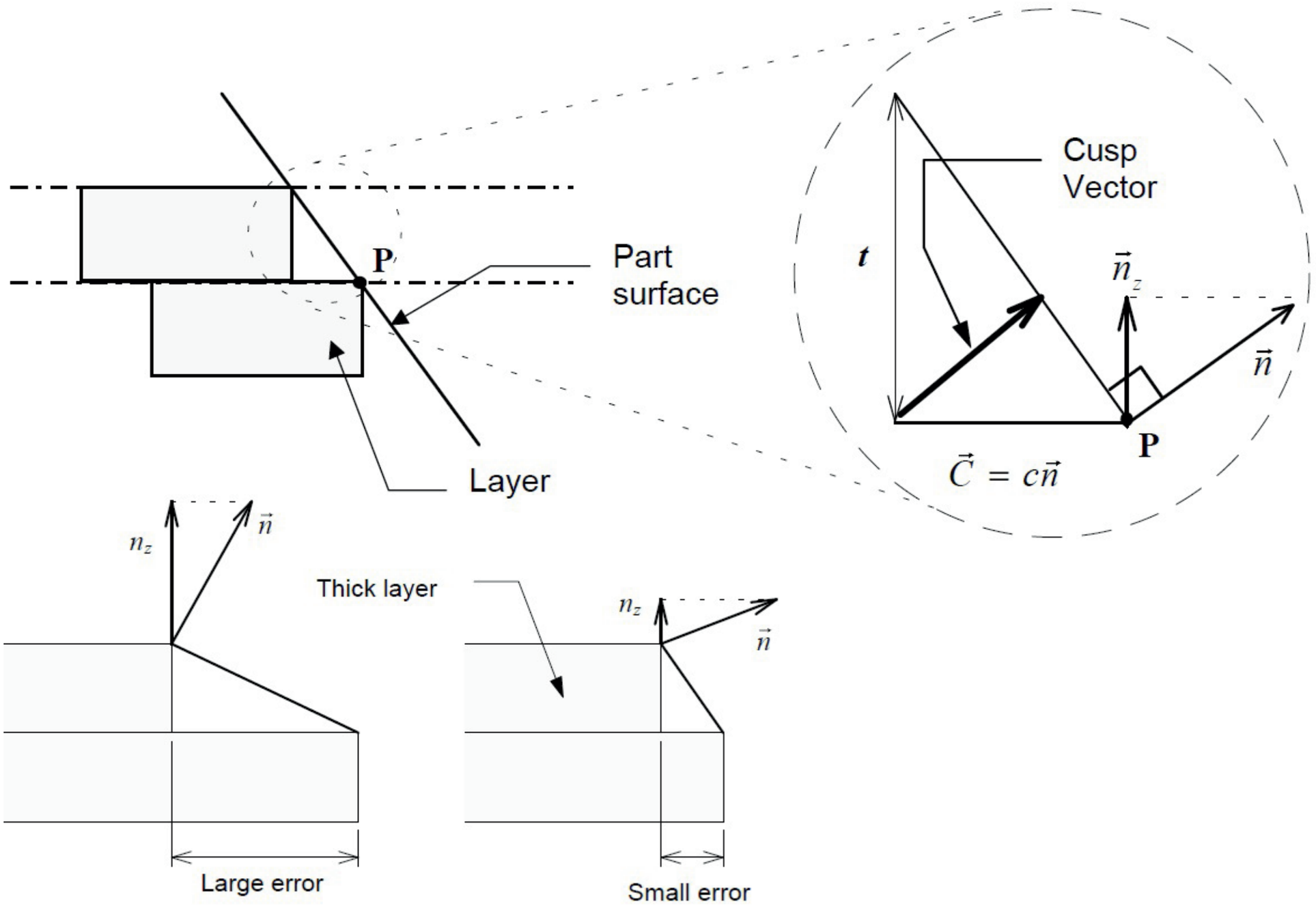
Verfahren (Ablauf)

Zweiter Teil: 2) Adaptives Slicing (exteriour)

- 3D-Drucker kennt seine minimal (th_{min}) und seine maximal (th_{max}) mögliche Schichtdicke
- der Parameter α, der den Grad der Verfeinerung angibt, wird eingeführt
- Einsatz der gespeicherten maximalen Normalen-vektoren

Verfahren (Ablauf)

Zweiter Teil: 2) Adaptives Slicing (exteriour)



Verfahren (Ablauf)

Zweiter Teil: 2) Adaptives Slicing (exteriour)

- Formel zur Berechnung von α:

$$r = \frac{Thick\ layer\ thickness}{Adaptive\ layer\ thickness} = \frac{th_{\max}}{\frac{C_{\max}}{n_z}} = n_z \frac{th_{\max}}{C_{\max}}$$

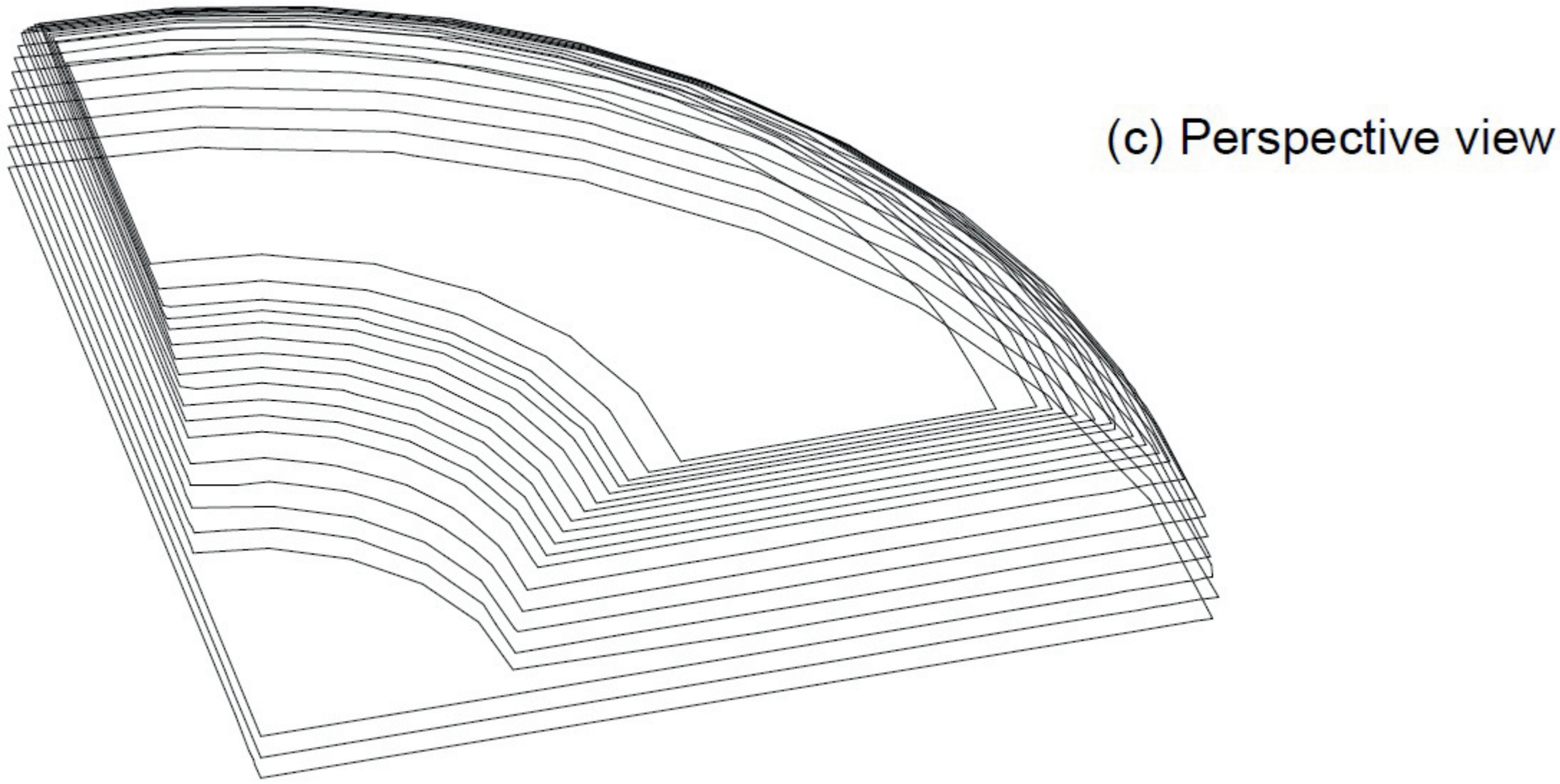
- C_{max} : größter erlaubter Fehler
- $\alpha \in [1, \alpha_{max}]$
- α ist dann der integer-Teil von r

Auswertung

Experiment:

- dasselbe 3D-Objekt auf 4 unterschiedliche Arten Slicen (normal mit dicken Schichten, normal mit dünnen Schichten, normales adaptives Slicen und interiour/exteriour), bei allen dieselbe .STL-Datei als Vorlage, und dann drucken
- Fokus auf Slicing-Genauigkeit (zum Vergleichen)

Auswertung



(c) Perspective view

Ergebnis (Oberflächen):

- abgesehen von “normal mit dicken Schichten” haben alle Objekte eine äquivalent gute Oberfläche

Auswertung

Ergebnis (Messwerte):

Method	Layer thickness	No. slices	Tool path length (mm)	Relative tool path length	Actual build time (h)	Relative build time	Matl. use (g) +/- 0.001	Relative Matl. use
(1) Uniform thick layers	0.0150"	25	9,843	40	0.5	36	5.558	100
(2) Uniform thin layers	0.0050"	75	24,710	100	1.4	100	5.312	96
(3) Adaptive slicing	0.0050" 0.0075" 0.0150"	38	17,885	72	0.85	61	5.295	95
(4) Shell/Interior	0.0050" 0.0075" 0.0150"	38	13,667	55	0.81	59	4.536	59

=> geringe Zeitersparnis im Vgl. zum *adaptive slicing*

=> hohe Materialersparnis

Auswertung

- Begründung: der *tool path*, der in dieser Arbeit verwendet wurde, ist zu komplex
- wenn man die relative *tool path*-Länge betrachtet erkennt man das Potential dieser Methode
- würde man das optimieren, soll eine Verringerung der Dauer auf 0,65 Stunden möglich sein

-> 30% langsamer als “dicke Schichten”

-> 115% schneller als “dünne Schichten”

-> 31% schneller als “normales Adaptives Slicen”

Ende

Quellen

- Hauptquelle:
  http://www-rp.me.vt.edu/bohn/projects/sabourin/thesis.pdf
- https://tams.informatik.uni-hamburg.de/publications/2014/adaptive_slicing_wasserfall.pdf