



Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

3D GANs

Vorstellung der Projektarbeit

14.01.2019

Jonas Harlacher & Alexander Melde



Intelligent Systems
Research Group

Inhaltsverzeichnis

1. Aufgabenstellung
2. Motivation
3. Grundlagen
4. Entwicklungsverlauf
5. Ergebnisse
6. Quick Start
7. Ausblick
8. Lessons Learned
9. Zusammenfassung
10. Quellenangaben

Aufgabenstellung

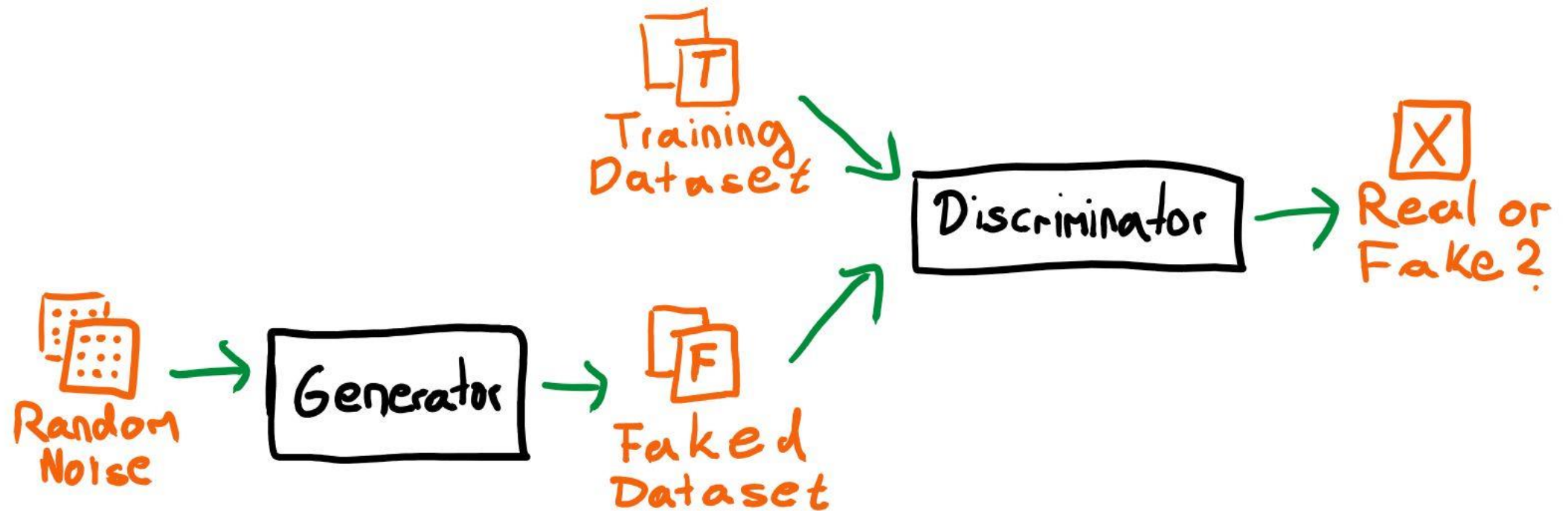
- GANs sind die neuen Deep Learning Netze, mit denen sich aktuell die Forschungsabteilungen und -Institute beschäftigen.
- Mit GANs lassen sich Klassifikatoren täuschen, sie können aber auch verwendet werden, um 3D Daten (Modelle) zu generieren.
- Evaluieren Sie den aktuellen Stand der Wissenschaft und Technik.
- Einstiegspunkt: <http://3dgan.csail.mit.edu/>

Motivation

- Die Forschungsgruppe Intelligent Systems Research Group (ISRG) der Hochschule Karlsruhe beschäftigt sich derzeit mit verschiedenen Projekten im Bereich 3D und Augmented Reality.
- Es könnte hilfreich sein, zwischen 3D-Objekte interpolieren zu können
- In dieser Projektarbeit soll geprüft werden, ob eine Interpolation mithilfe von 3D GANs möglich ist.

Grundlagen

Generative Adversarial Networks (GANs)



nach [WI18]

Entwicklungsverlauf

① Auswahl der Entwicklungsumgebung



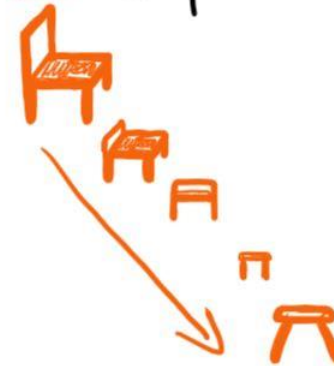
② Einrichtung der Entwicklungsumgebung



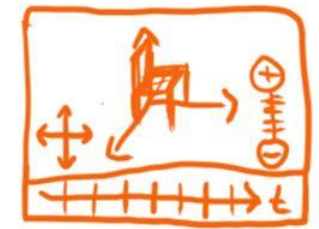
③ Generierung von 3D-Modellen



④ Interpolation



⑤ Visualisierung



Entwicklungsverlauf

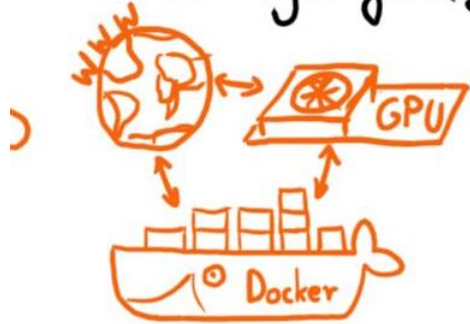
① Auswahl der
Entwicklungsumgebung



- VM hat zu wenig Leistung
 - GPU erforderlich
 - Linux-System ist für Torch Voraussetzung
- Rechner im LfM mit 1080Ti

Entwicklungsverlauf

② Einrichtung der Entwicklungsumgebung



- Installation von Docker um Adminrechte-Einschränkung zu umgehen
- Eigene Skripte um Proxy durchzureichen
- Installation von CUDA für GPU-Support
- Installation von Torch, Tensorflow, ... nachvollziehbar innerhalb der Dockerfiles

Entwicklungsverlauf

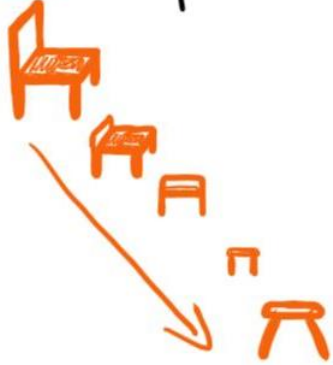
③ Generierung von 3D-Modellen



- Drei funktionierende Ansätze:
 - *3dgan-release* (Torch) [WU16]
 - *3D-IWGAN* (Tensorflow) [SM17]
 - *coms-project* (Tensorflow) [RP17]
- Trainieren eines 3D-GANs (Diskriminator & Generator) und Test mit zufälligen Eingabevektoren (Z-Vektoren) führt zu zufällig generierten 3D-Modellen

Entwicklungsverlauf

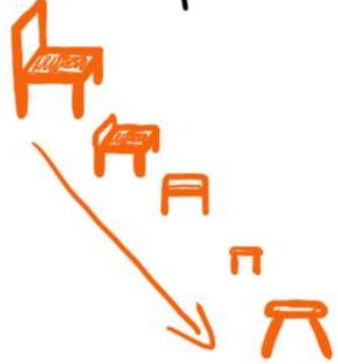
④ Interpolation



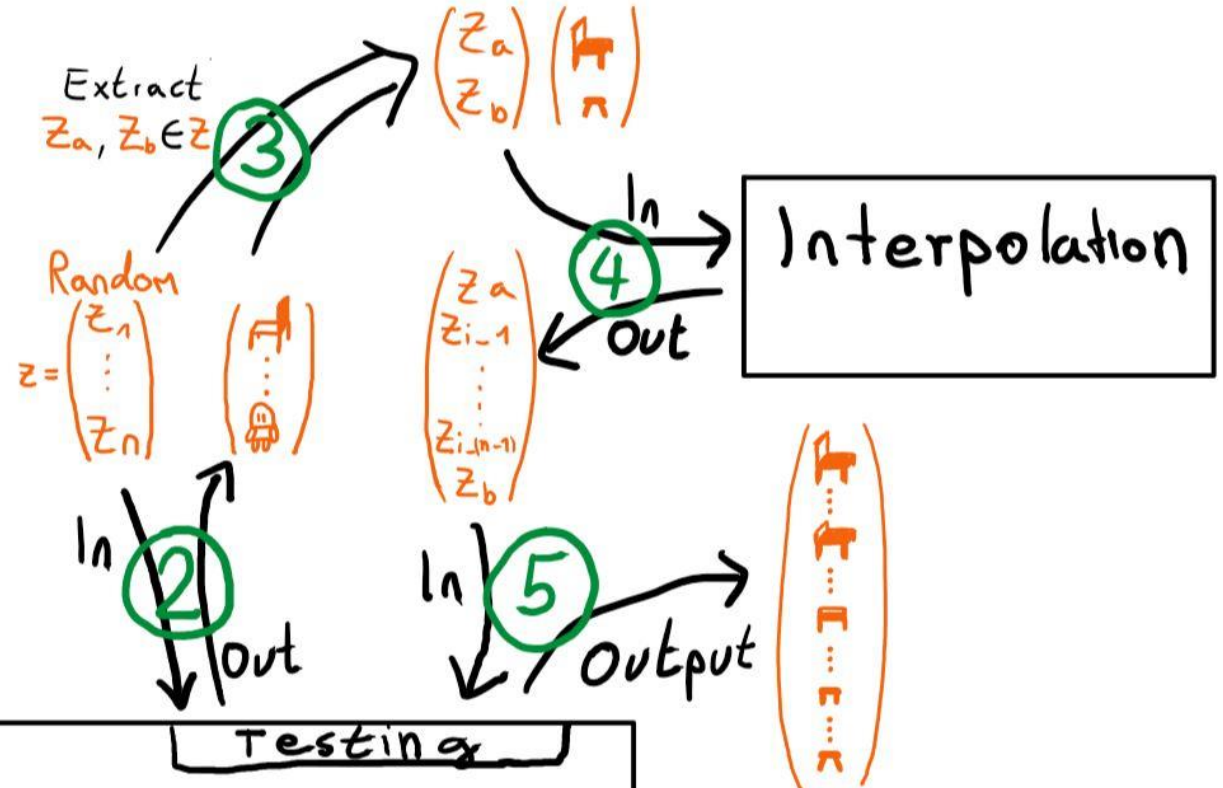
- Ansatz 1:
 - Generieren mit *3D-WGAN* und Interpolieren mit *coms-project*
 - → Fehlgeschlagen, interpolierte 3D-Modelle sind keine Zwischenschritte sondern einfach weitere zufällig generierte 3D-Modelle.
- Ansatz 2:
 - Generieren und Interpolieren mit *coms-project*
 - → Erfolgreich!

Entwicklungsverlauf

④ Interpolation

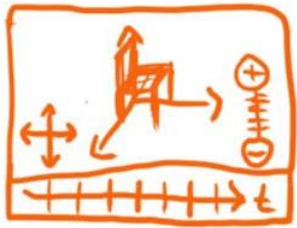


① Input



Entwicklungsverlauf

⑤ Visualisierung



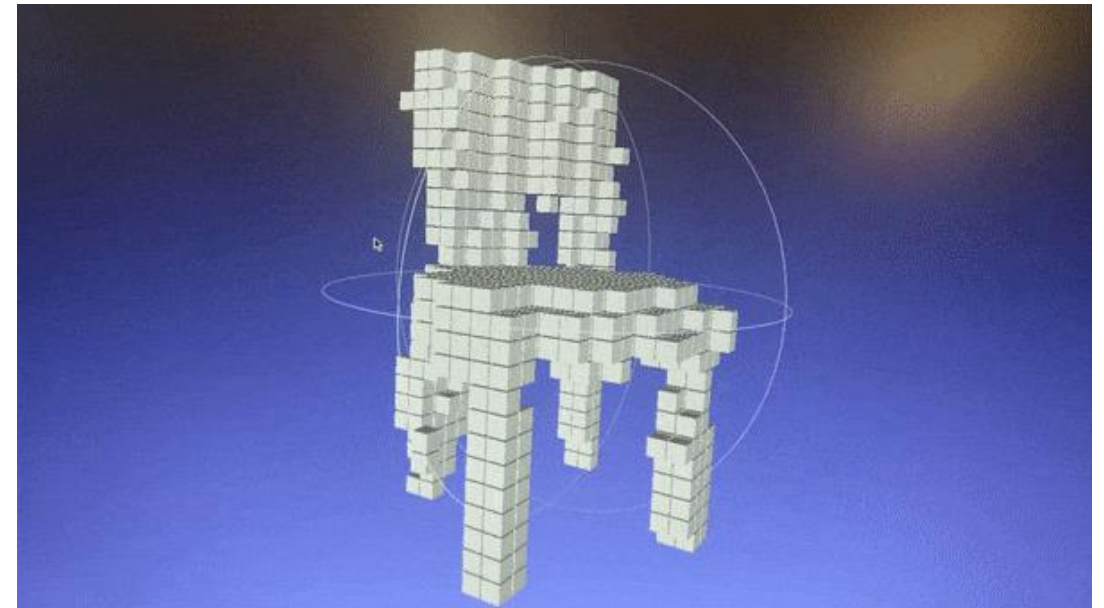
- Zu jedem Ansatz mindestens eine Methode zur Visualisierung getestet
 - Funktionsaufrufe siehe Dokumentation
- Selbst entwickelt wurden zwei Skripte zur Veranschaulichung der Interpolation:
 - Automatisiertes erstellen von Screenshots des 3D-Viewers MeshLab für animierte Gifs
 - Automatisiertes Ein- und Ausblenden von 3D-Modellen in Blender zum Erstellen von Videos

Ergebnisse

Generierung



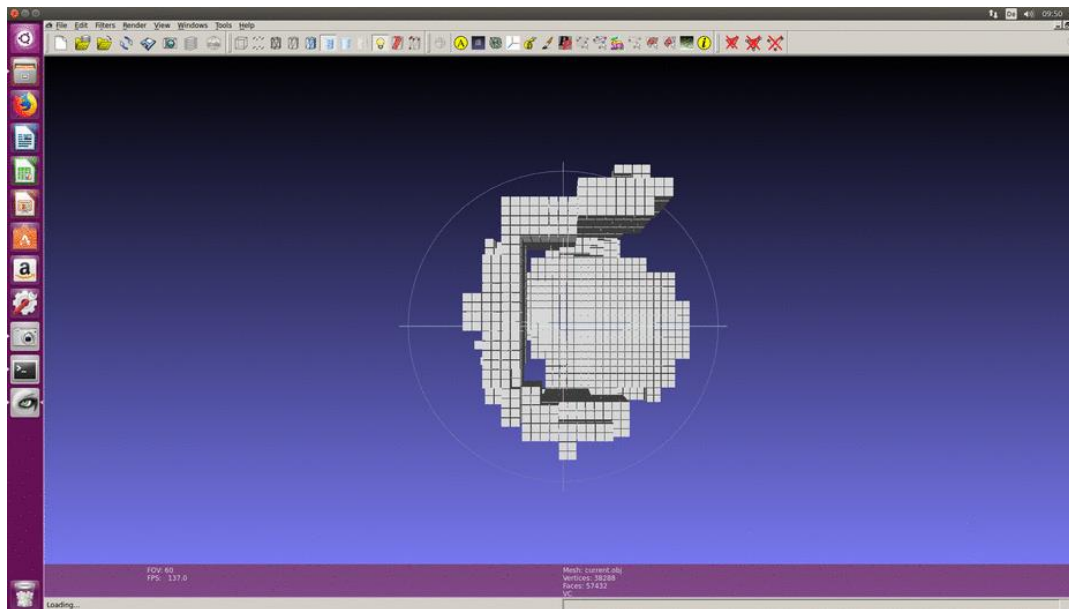
Generiert und Visualisiert mit 3dgan-release



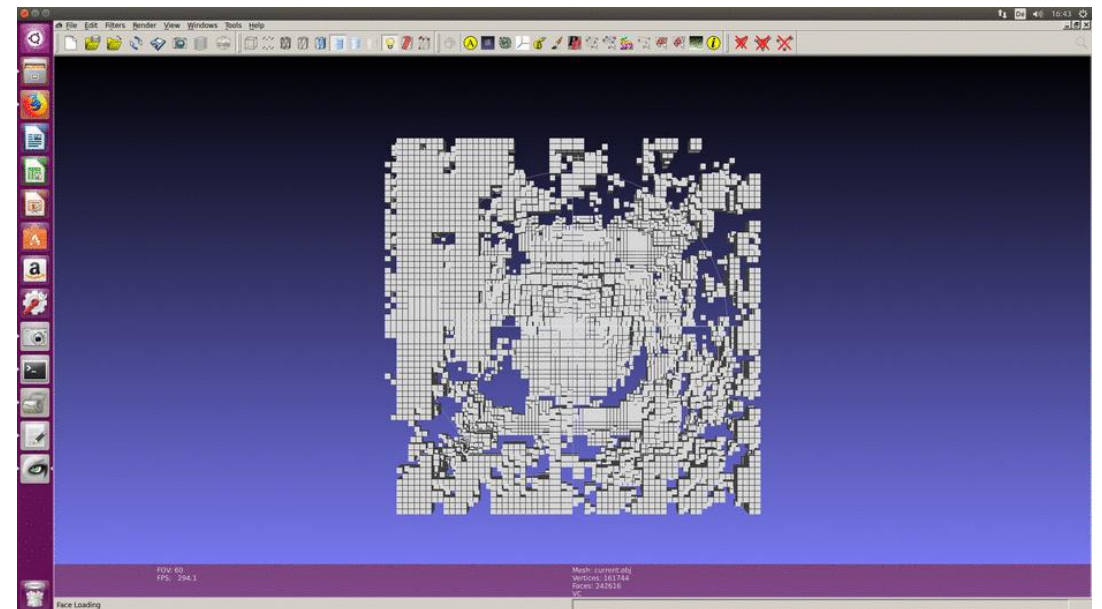
Generiert mit 3D-IWGAN, visualisiert mit MeshLab

Ergebnisse

Interpolation



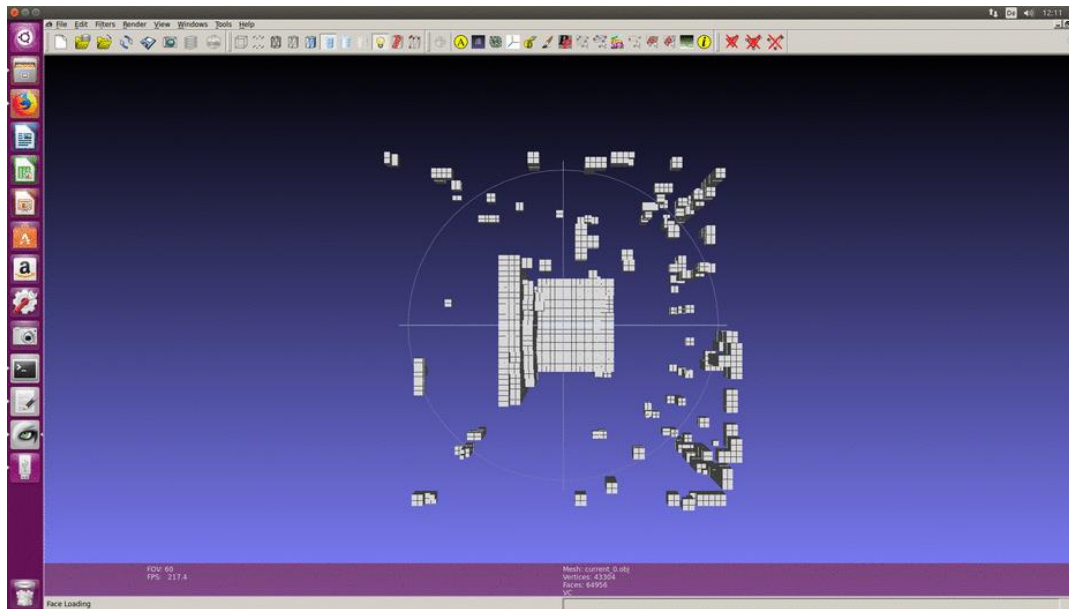
1912chair02_14600_intpol_2_4__interpolated_results.npy



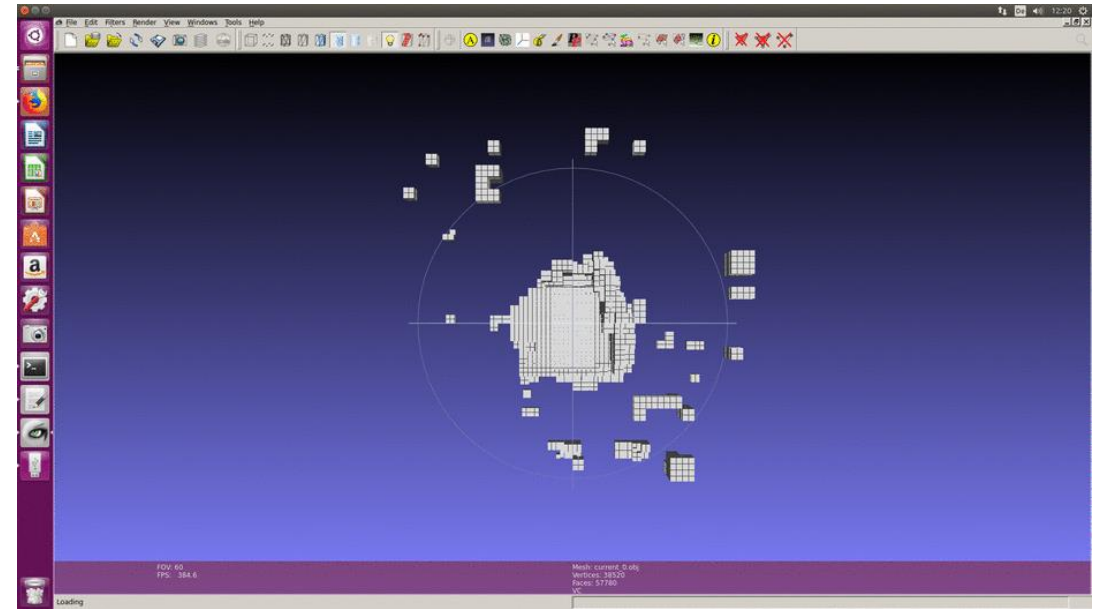
lfmTest4200_intpol_2_4__interpolated_results.npy

Ergebnisse

Interpolation



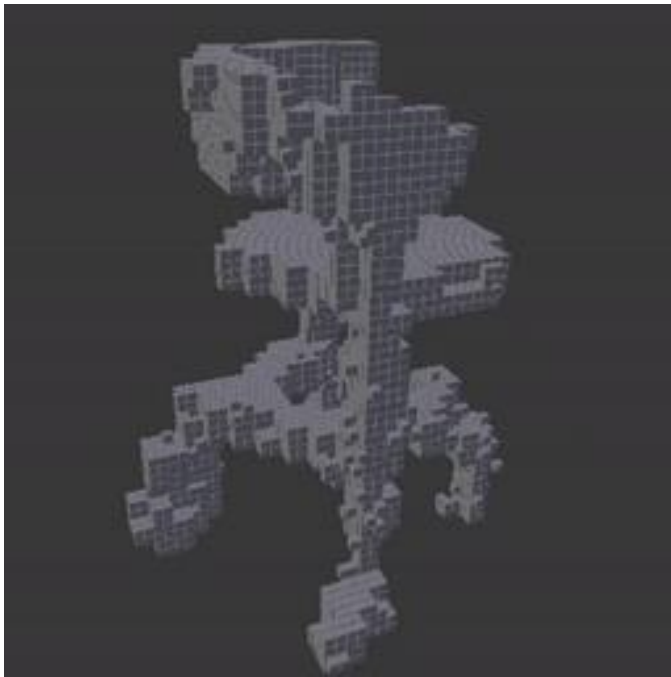
2112chair01_36800_intpol_2_5__interpolated_results.npy



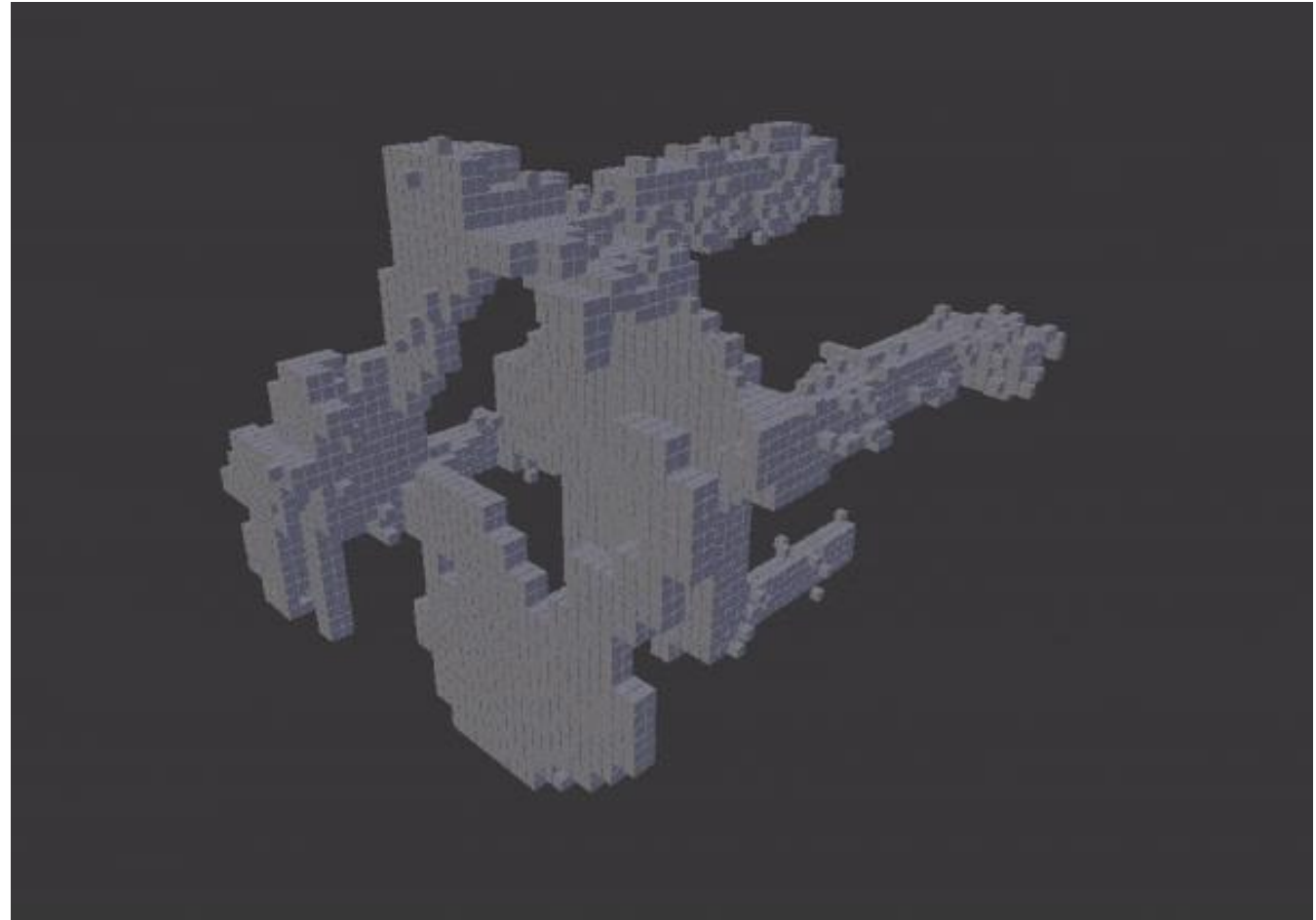
2012chair01_36000_intpol_9_10__interpolated_results.npy

Ergebnisse

Interpolation



generiert und interpoliert mit coms-project,
visualisiert mit eigenem Blender-Skript



Quick Start

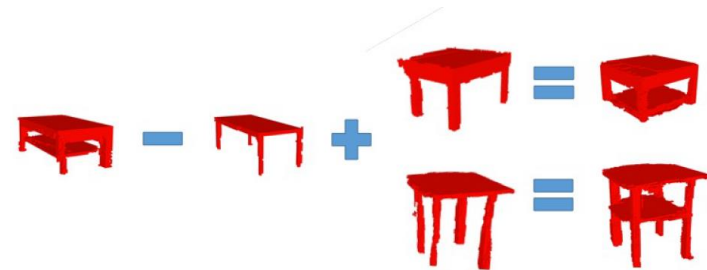
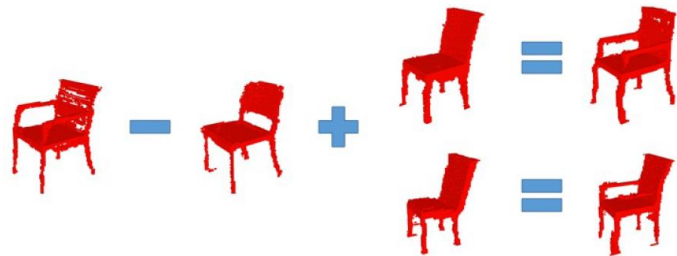
- 3D Chair Generation & Interpolation (coms-project) [RP17]
- Generiert beispielhafte Stühle, interpoliert zwischen zwei dieser Stühle, speichert die interpolierten 3D-Objekte und öffnet ein Programm zum Betrachten dieser.

```
cd lfm_coms
source ./prepare_host
./build.sh
./run.sh
./include/interpolation.sh
```

```
# choose subproject
# configure proxy
# build docker container
# run docker container
# start interpolation routine
```

Ausblick

- Interpolation zwischen 3D-Objekten verschiedener Klassen
- Ursache für Artefaktbildung bei Interpolation mancher Z-Vektor-Paare untersuchen
- Anwendung von Arithmetik auf Z-Vektoren testen



[WU16]

Lessons Learned

- Interpolation zwischen 3D-Objekten mithilfe von GANs ist möglich.
- Bestehende Quelltexte müssen für die eigene Entwicklungsumgebung in den meisten Fällen angepasst werden.
- Das entwickelte Dockerfile *lfm_coms* ermöglicht einen einfachen plattformübergreifenden Interpolations-Test mit wenigen Befehlen.

Zusammenfassung

- Entwicklungsumgebung geschaffen für Hochschul-PCs mit Proxy
- Grundlegendes Verständnis 3D-GAN geschaffen
- Projektverlauf und Ergebnisse dokumentiert
 - Auswahl und Einrichtung der Entwicklungsumgebung
 - Generierung, Interpolation und Visualisierung von 3D-Modellen
 - drei Ansätze: *3dgan-release* [WU16], *3D-IWGAN* [SM17], *coms-project* [RP17]
- Schritt-für-Schritt-Anleitung und Skript zur Interpolation erstellt
 - → mittels *coms-project* [RP17]
- Zwei Visualisierungsskripte (MeshLab GIF und Blender Video)

Quellenangaben

- Grundlagen zu Neuronalen Netzen und GANs
 - [WI18] *Wiegand* (2018) „Eine Einführung in Generative Adversarial Network(GAN)“ ([Paper](#))
- 3D-GANs
 - [WU16] *Wu, Zhang, Xue, Freeman, Tenenbaum* (2016) „Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling“ in „Advances In Neural Information Processing Systems“ (p. 82-90) ([Homepage](#), [Paper](#), [GitHub](#))
 - [SM17] *Smith, Meger* (2017) „Improved Adversarial Systems for 3D Object Generation and Reconstruction“ ([Paper](#), [GitHub](#))
 - [RP17] *rp2707, Greg K (wasd12345)* (2017) „coms4995-project“ ([GitHub](#))