



# netidee

PROJEKTE

## CADMeshConverter Entwicklerdokumentation

Dokumentation | Call 13 | Projekt ID 3511

Lizenz: CC-BY-SA

## Inhalt

1	Über das Projekt.....	4
2	GITHUB / Solution.....	5
3	Software Architektur.....	6
3.1	Core.....	6
3.2	Cloud-Service / Backend-Service .....	6
3.3	Client – CMC WebApplication .....	10
3.4	Android Smartphone APP – Showcase der API .....	11
4	Automatischer Workflow .....	12
4.1	Import funktionalität .....	12
4.2	Ausbessern & Bereinigen des CAD Modells .....	12
4.3	Normalen & Orientierung.....	12
4.4	Remeshing, Simplifizierung & Rekonstruktion .....	12
4.5	Quadratic Edge Collapse Decimation.....	12
4.6	Export Funktionalität.....	12
4.7	Zusätzliche Funktionalität .....	12
5	Installation.....	12
6	Konfiguration.....	13
6.1	CMC Cloud - RESTAPI .....	13
6.2	CMC Web.....	13
6.3	CMC Smartphone APP.....	14
7	Technische Eigenschaften.....	14
8	Unterstützte Formate.....	15
8.1	Import Formate .....	15
8.2	Export Formate.....	15
9	Open Source Software & Bibliotheken.....	15
9.1	Fast Quadric Mesh Simplification .....	15
9.2	Meshlab .....	16
9.3	JMeshLib.....	16
9.4	PyMesh.....	16

<u>9.5</u>	<u>OpenFlipper.....</u>	<u>17</u>
<u>9.6</u>	<u>Blender .....</u>	<u>17</u>

# 1 Über das Projekt

Die Umsetzung von Augmented Reality (AR) -Projekten stellt Industriepartner bei der Integration von CAD-Modellen immer wieder vor die gleichen Hürden. Die Erstellung von 3-D-Modellen im Bereich mobiler Endgeräte ist sehr zeit- und kostenintensiv: Sie erfolgt manuell, wodurch häufig zusätzliche Funktionalitäten hinfällig werden. Der Wunsch nach einem automatisierten Workflow wird immer größer, schließlich musste sich Entwickler schon viel mit Konvertierungen und Konvertierungstools beschäftigen und hatte somit weniger Zeit für die spannenden Dinge der AR-Entwicklung.

Der CADMeshConverter ist ein Open-Source-Tool, mit dem hochdetaillierte CAD-Modelle in leichte Modelle für die AR / VR / MR-Visualisierung konvertiert werden können.

Das Projekt beinhaltet folgende Bestandteile:

- Dem Core-Modul, dass den Konvertierungsprozess durchführt
- Eine REST API die auf einem Server ausgeführt werden muss und den Konvertierungsprozess kapselt (Swagger)
- Eine Webapplikation, mit der die Konvertierung und auch benutzerdefinierten Skripte getestet werden kann (einige Skripte sind bereits enthalten)
- Eine Android Applikation um die Funktionalität zu demonstrieren

## 2 GITHUB / Solution

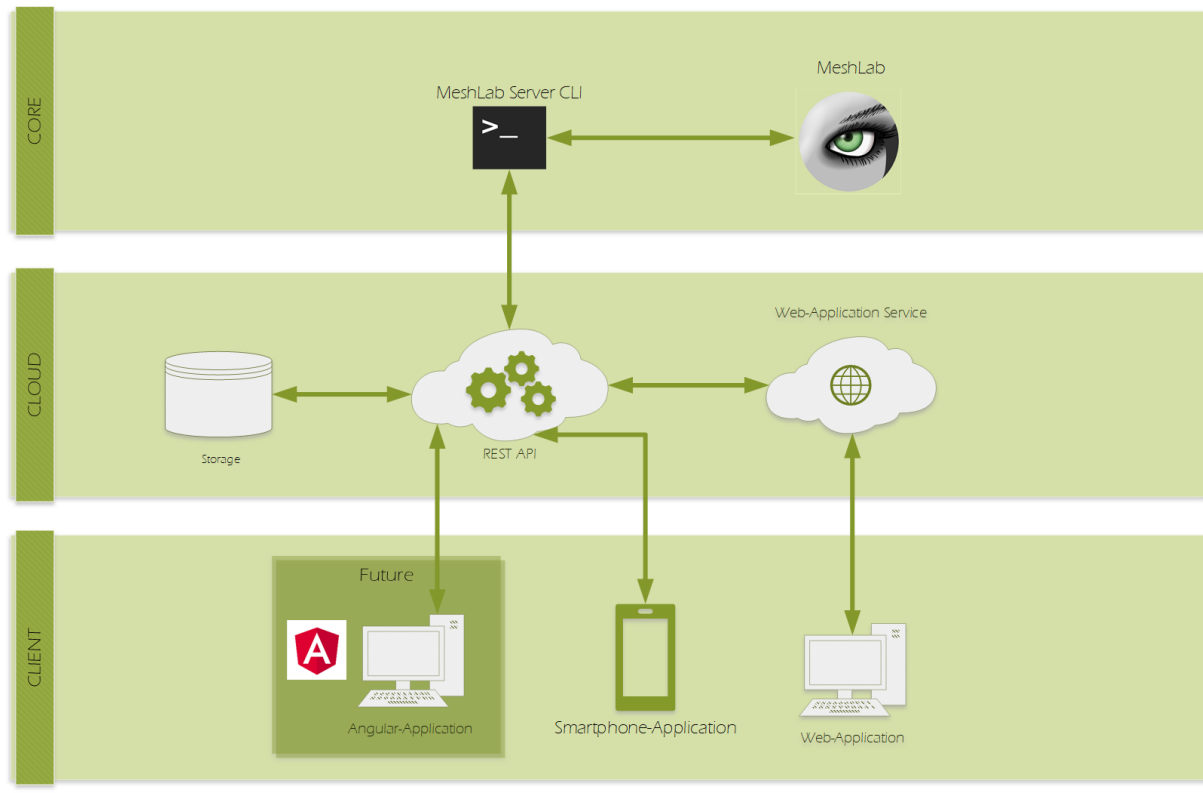
Die gesamte Software-Lösung steht auf GitHub unter folgendem Link zur Verfügung

<https://github.com/FotecGmbH/CADMeshConverter>

Nähere Informationen finden sich im Repository selbst.

[TODO: Screenshot]

## 3 Software Architektur



### 3.1 Core

#### MeshLab Server CLI

Über die MeshLab Server CLI können Jobs gestartet werden.

Folgende Parameter werden übergeben:

- Datei
- Eingabeformat
- Skript
- Ausgabeformat

#### MeshLab

MeshLab führt die Operationen wie Remeshing, Simplification und Reconstruction durch und liefert das Ergebnis in dem angegebenen Ausgabeformat.

### 3.2 Cloud-Service / Backend-Service

#### Storage / Installationsordner

Im Installationsordner sind die Skripte für MeshLab-Operationen gespeichert.

Name	Änderungsdatum	Typ	Größe
ClusteringDecimation.mlx	13.09.2019 15:10	MATLAB Live Script	1 KB
HiddenRemoval.mlx	18.09.2019 09:36	MATLAB Live Script	3 KB
NormalizeVertexNormals.mlx	19.09.2019 13:00	MATLAB Live Script	1 KB
QuadricEdgeCollapseDecimation.mlx	18.09.2019 09:35	MATLAB Live Script	4 KB
RecomputeVertexNormals.mlx	19.09.2019 12:57	MATLAB Live Script	1 KB
RemoveDuplicateVertices.mlx	19.09.2019 12:58	MATLAB Live Script	1 KB
RemoveUnreferencedVertices.mlx	19.09.2019 12:58	MATLAB Live Script	1 KB
RepairSimplifyAndMeasureHauDl.mlx	13.09.2019 15:27	MATLAB Live Script	7 KB
ScreenedPoissonSurfaceReconstruction.mlx	19.09.2019 12:58	MATLAB Live Script	1 KB

In jedem Skript sind die Parameter, die im Skript für die Operationen verwendet werden, definiert. Die Definition der Skriptdatei entspricht bis auf drei Ausnahmen dem Meshlab Standard<sup>1</sup>.

- Name ... Name der im Web GUI Dropdown verwendet wird
- hiddenFilter ... Filter Gruppe wird im Experten Modus nicht angezeigt
- hiddenParam ... Parameter wird im Experten Modus nicht angezeigt

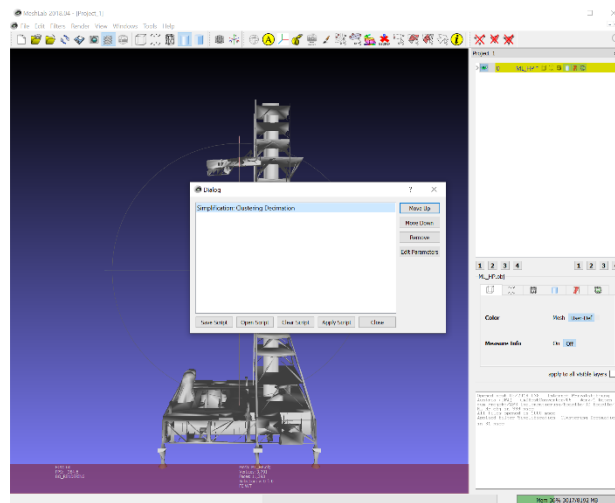
```
<!DOCTYPE FilterScript>
<FilterScript name="RepairSimplifyAndMeasureHauasdorffDistance">
  <filter name="Duplicate Current layer"/>
  <filter name="Remove Unreferenced Vertices"/>
  <filter name="Re-Compute Vertex Normals">
    <Param tooltip="" isxmlparam="0" value="0" enum_cardinality="4" description="Weighting
Mode:" enum_val1="By Angle" enum_val2="By Area" type="RichEnum" enum_val3="As defined by N.
Max" name="weightMode" enum_val0="None (avg)"/>
  </filter>
  <filter name="Normalize Vertex Normals"/>
  <filter name="Normalize Face Normals"/>
  <filter name="Simplification: Quadric Edge Collapse Decimation" hiddenFilter="true">
    <Param tooltip="If non zero, this parameter specifies the desired final size of the mesh as
a percentage of the initial size." isxmlparam="0" value="0.5" description="Percentage
reduction (0..1)" type="RichFloat" name="TargetPerc" hiddenParam="true" />
    <Param tooltip="Quality threshold for penalizing bad shaped faces.<br>The value is in the
range [0..1]&#xa; 0 accept any kind of face (no penalties),&#xa; 0.5 penalize faces with
quality < 0.5, proportionally to their shape&#xa;" isxmlparam="0" value="0.3"
description="Quality threshold" type="RichFloat" name="QualityThr"/>
    .....
  </filter>
  <filter name="Delete Current Mesh"/>
</FilterScript>
```

<sup>1</sup> MeshLab Standard: <http://www.meshlab.net>

### Skripterstellung via MeshLab

Die grafische Benutzeroberfläche von MeshLab protokolliert alle durchgeführten Operationen mit. Diese Tatsache kann dazu genutzt werden, Filterscripts in der GUI zu erstellen. Dies geschieht folgendermaßen:

- Ein Modell in MeshLab öffnen
- Die Operationen die in einem Filterskript vorhanden sein sollen auf das Modell anwenden (z.B.: Clustering Decimation)
- Den Menüpunkt: „Filters“ -> „Show current filterscript“ auswählen.



- Es wird eine Liste mit den durchgeführten Operationen angezeigt. Die Reihenfolge der einzelnen Punkte sowie deren Parameter können geändert werden.
- Sobald alle Änderungen vorgenommen wurden, kann das Script gesichert werden.

Um so erzeugte Scripts für den CADMeshConverter zu adaptieren, können bzw. müssen die zuvor erwähnten XML Parameter verwendet werden. Dies kann mit Hilfe eines beliebigen Text Editors geschehen.



## Exchange Klassen

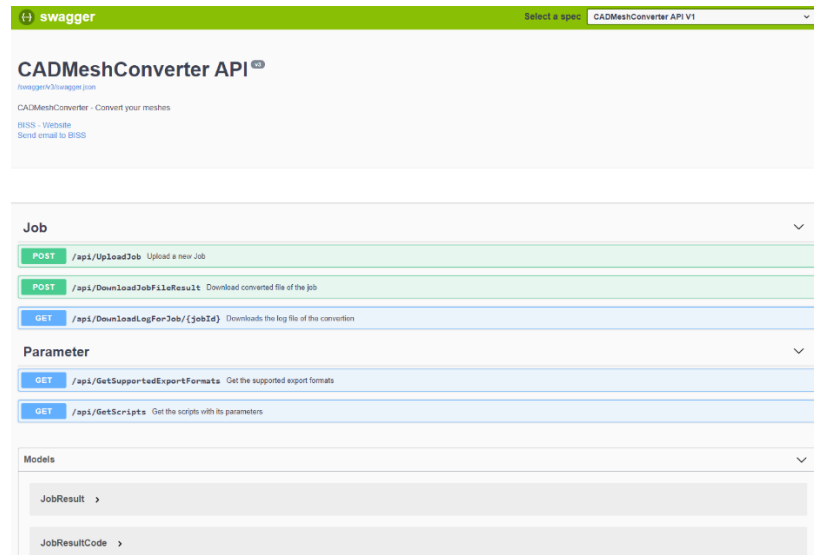
Diese Klassen dienen den Datenaustausch zwischen einem Client und der Rest API.

- Exchange
  - Enum
    - C# Enums.cs
  - Models
    - C# FilterScript.cs
    - C# FilterScriptExt.cs
    - C# JobResult.cs
    - C# JobResultCode.cs
    - C# MeshInfo.cs
    - C# Script.cs
    - C# ScriptParameter.cs
    - C# ScriptParameterFilter.cs
    - C# SupportedExportFormat.cs
- Exchange.csproj

## RESTAPI – meshapi.fotec.at

Über die REST API (Swagger) können MeshLab Jobs gestartet werden. Die Dokumentation ist dem Swagger Standard entsprechend leicht zu finden:

<http://meshapi.fotec.at:8091/swagger/index.html>



- RESTApi
  - Controllers
  - Helper
  - Properties
  - SwaggerOperations
  - () appsettings.Development.json
  - () appsettings.json
  - CMC Cloud.csproj
  - <? CMC Cloud.xml
  - () configuration.json
  - C# Constants.cs
  - output.zip
  - C# Program.cs
  - <? RESTApi.xml
  - C# Startup.cs
  - <? web.config

Folgende Parameter müssen für einen Job angegeben werden:

- Skript
- Parameter für das Skript
- Eingabedatei
- Format der Eingabedatei
- Ausgabeformat

Folgende Informationen können über die REST-API abgefragt werden:

- Liste von unterstützten Eingabeformaten
- Liste von unterstützte Ausgabeformaten
- Liste an Skripten inklusive der verwendbaren Parameter

Die REST-API nimmt über eine Schnittstelle alle Daten entgegen, ersetzt die angegebenen Parameter mit den Parametern im Skript und startet anschließend den MeshLab Job über die MeshLab Server CLI. Ist der Job abgeschlossen liefert die REST-API das Ergebnis (Ausgabedatei) zurück.

### Web-Application Service

Das Web-Application Service holt sich alle Verfügbaren Informationen und Daten von der REST-API und bereitet diese für die Darstellung auf einer Web-Applikation vor. Ruft ein Client das Web-Application Service auf wird diesem die View zurückgeliefert um folgende Daten anzugeben:

- Skript
- Parameter
- Datei
- Format der Eingabedatei
- Ausgabeformat

Hat ein Client alle Daten angegeben und übermittelt wird über die REST API der Job gestartet und dem Client das Resultat zur Verfügung gestellt.

```

Services
  CMC Core.csproj
  C# FilterScriptService.cs
  C# MeshLabService.cs
  
```

## 3.3 Client – CMC WebApplication

### Web-Application

Durch Aufruf der URL des Web-Application Services wird dem Client eine grafische Benutzeroberfläche angezeigt, in dieser er folgende Daten eingeben kann:

- **Datei**  
Kann von dem lokalen Speicher des Clients ausgewählt werden
- **Format der Eingabedatei**  
Muss von der zur Verfügung gestellten Liste ausgewählt werden
- **Skript**  
Muss von der zur Verfügung gestellten Liste ausgewählt werden
- **Parameter des ausgewählten Skriptes**  
Müssen eingegeben werden (Werden nach Auswahl des Skriptes sichtbar)

```

WebApplication
  Controllers
  Helper
  Models
  Properties
  ServiceAccess
  Views
  wwwroot
  () appsettings.Development.json
  () appsettings.json
  CMC Web.csproj
  () libman.json
  C# Program.cs
  C# Startup.cs
  </> web.config
  
```

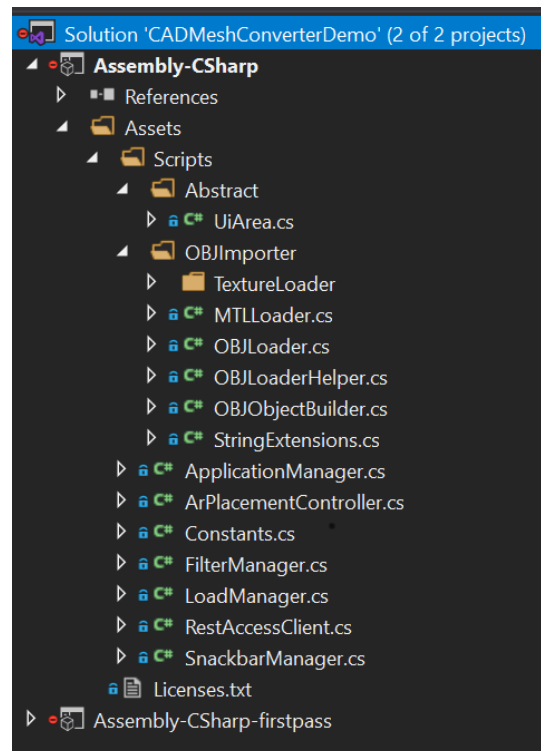
- **Ausgabeformat**

Muss von der zur Verfügung gestellten Liste ausgewählt werden

Hat der Client alle Daten angegeben und ausgefüllt kann er diese mit einem Bestätigungs-Button an das Web-Application Service übermitteln. Der Job wird anschließend am Server verarbeitet. Sobald der Job abgeschlossen ist, wird dem Client benachrichtigt und dieser kann die fertige Datei anschließend herunterladen.

### 3.4 Android Smartphone APP – Showcase der API

Bei der Android App handelt es sich um eine Augmented Reality App für Android, die die Anbindung an die REST Schnittstelle des CADMeshConverters demonstriert. Implementiert wurde sie mit AR Foundation und Unity 2019.1.10. Die Zugriffe auf die REST Schnittstelle erfolgen über die „RestAccessClient.cs“ Klasse.



- **Rest Schnittstelle**

Die URL zur Rest schnittstelle kann in der „Constants.cs“ konfiguriert werden.

- **Funktionen**

Die Anwendung unterstützt nur die Reduktion eines Modells, um weitere Funktionen verwenden zu können, muss ein Filterscript Parser implementiert werden.

- **Eingabeformate**

Die unterstützten Eingabeformate sind von der am Server verwendeten MeshLab Version abhängig. Sie können in der „Constants.cs“ parametrisiert werden.

- **Ausgabeformat**

Um Modelle in der App reduzieren und verwenden zu können, muss das Ausgabeformat „OBJ“ verwendet werden. Nur dieses Format kann zur Laufzeit in der App importiert werden.

- **Ausgabedatei**

Im Demonstrator wird immer nur ein reduziertes Modell abgespeichert. Dies bedeutet, dass Reduktionsvorgänge ein bereits vorhandenes Reduktionsmodell überschreiben.

- **Verwendete Plugins**

Die verwendeten Plugins für Unity sind in der Datei „Licenses.txt“ beschrieben

## 4 Automatischer Workflow

Der automatische Workflow – sprich die automatische Konvertierung eines CAD Modells zu einem mobilfreundlichen 3D Modell – inkludiert unter anderem folgende Features:

### 4.1 Import funktionalität

- Import von STL
- Import von OBJ
- Import von div. anderen Formaten

### 4.2 Ausbessern & Bereinigen des CAD Modells

- Entfernen von unreferenzierten Vertizes
- Entfernen von duplikaten Vertizes

### 4.3 Normalen & Orientierung

- Neuberechnung von Vertexnormalen
- Normalisieren von Vertexnormalen

### 4.4 Remeshing, Simplifizierung & Rekonstruktion

- Screened Poisson Surface Reconstruction (für 3D Scans, damit diese “wasserdicht” gemacht werden, sprich eine durchgängige Oberfläche haben)

### 4.5 Quadratic Edge Collapse Decimation

- Simplifizierung eines Modells, damit weniger Vertizes und Faces vorhanden sind

### 4.6 Export Funktionalität

- Export des Meshes als OBJ
- Export des Meshes als glTF

### 4.7 Zusätzliche Funktionalität

- Hidden Removal (Entfernen von nicht sichtbaren Flächen)

## 5 Installation

- MeshLab installieren
- Lokales Arbeitsverzeichnis erstellen (z.B. C:\Mesh\Work)
- Lokales Script Verzeichnis erstellen (z.B. C:\Mesh\Scripts)
- Visual Studio installieren
- CADMeshConverter Solution öffnen
- Verzeichnisse in CMC Cloud konfigurieren (siehe 6.1)
- Scripts in Script Ordner kopieren (siehe 6.1)

Los gehts!

## 6 Konfiguration

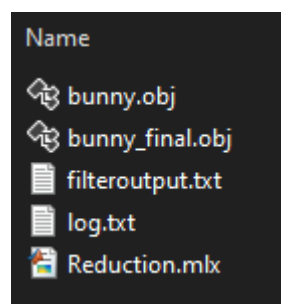
### 6.1 CMC Cloud - RESTAPI

Die Datei appsettings.json befindet sich im "CMC Cloud" Projekt.

*Note: "MeshLabDirectory" ist das Installationsverzeichnis von MeshLab\**

```
{
  "AppSettings": {
    "WorkingDirectory": "C:\\TEMP\\Mesh\\Work\\",
    "ScriptDirectory": "C:\\TEMP\\Mesh\\Scripts\\",
    "MeshLabDirectory": "C:\\TEMP\\Mesh\\MeshLab\\",
    "UploadDirectoryName": "Upload",
    "ResultFileNameEnding": "_final",
    "FilterOutputFilename": "filteroutput"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

WorkingDirectory	temporäres Verzeichnis zum Prozessieren der Daten
ScriptDirectory	Skript Installationsverzeichnis
MeshLabDirectory	Meshlab Installationsverzeichnis
UploadDirectoryName	Upload Verzeichnis (relativer Pfad)
ResultFileNameEnding	Postfix zum original Namen
FilterOutputFilename	Informationsdatei



### 6.2 CMC Web

appsettings.json befindet sich im "CMC Web" Projekt.

```
{
  "AppSettings": {
```

```
"RestApiUrl": "http://meshapi.fotec.at:8091"
},
"Logging": {
  "LogLevel": {
    "Default": "Warning"
  }
},
"AllowedHosts": "*"
}
```

RestApiUrl	Protokoll:Host:Port der verwendeten Rest Schnittstelle
LogLevel	Setzt Trace, Information, Warning, Error für die Verwendung im Consolen Modus

### 6.3 CMC Smartphone APP

Die einzig notwendige Konfiguration in der Smartphone APP findet sich in der ApplicationManager.cs in der Methode Start().

```
void Start()
{
    _controller = FindObjectOfType<ArPlacementController>();
    _client = new RestAccessClient("http://meshapi.fotec.at:8091", new HttpCli
ent());
    _persistentPath = Application.persistentDataPath;
    _snackBar = FindObjectOfType<SnackBarManager>();
    FileBrowser.SetFilters(true, Constants.InputFormats);
}
```

RestAccessClient	Protokoll:Host:Port der verwendeten Rest Schnittstelle
------------------	--

## 7 Technische Eigenschaften

Der CADMeshconverter ist mit allen gängigen Browsern bedienbar und unter folgender URL erreichbar: <http://meshweb.fotec.at/MeshConverter>

## 8 Unterstützte Formate

### 8.1 Import Formate

- 3D Studio File Format (\*.3ds)
- Stanford Polygon File Format (\*.ply)
- STL File Format (\*.stl)
- Alias Wavefront Object (\*.obj)
- Quad Object (\*.qobj)
- Object File Format (\*.off)
- PTX File Format (\*.ptx)
- VCG Dump File (\*.vmi)
- FBX Autodesk Interchange Format (\*.fbx)
- Breuckmann File Format (\*.bre)
- Collada File Format (\*.dae)
- OpenCTM compressed format (\*.ctm)
- Expe's point set - binary (\*.pts)
- Expe's point set - ascii (\*.apts)
- XYZ Point Cloud - with and without normal (\*.xyz)
- Protein Data Bank (\*.pdb)
- TRI - photogrammetric reconstructions (\*.tri)
- ASC - ascii triplets of points (\*.asc)
- TXT - generic ASCII point list (\*.txt)
- X3D File Format - XML encoding (\*.x3d)
- X3D File Format - VRML encoding (\*.x3d)
- VRML 2.0 File Format (\*.wrl)

### 8.2 Export Formate

- Alias Wavefront Objekt (\*.obj)
- X3D File Format - XML encoding (\*.x3d)
- 3D Studio File Format (\*.3ds)
- Javascript JSON (\*.json)
- GL Transmission Format 2.0 (\*.gltf)

## 9 Open Source Software & Bibliotheken

### 9.1 Fast Quadric Mesh Simplification

Hierbei handelt es sich um eine in C++ geschriebene Programmbibliothek welche die Reduzierung von Meshes mittels der „Quadric based edge collapse“ Methode ermöglicht. Das Hauptaugenmerk des Entwicklers liegt dabei auf der Verarbeitungsgeschwindigkeit, worunter jedoch die Qualität des Resultats leiden kann.

Programmiersprache	C++
Lizenz	MIT
Unterstützte Import Formate	OBJ
Unterstützte Export Formate	OBJ

Vereinfachungsmethoden	Ja
Reparaturmethoden	Nein
Hierarchie Support	Nein
LOD Generierung	Nein
Texturoperationen	Nein

## 9.2 Meshlab

Ist ein komplettes Softwarepaket zur Be- und Verarbeitung von Meshes sowie für deren Erzeugung aus Punktwolken. Das Paket ist unterteilt in eine GUI und in ein Commandline Tool. Die GUI stellt sowohl eine interaktive Bearbeitungsmöglichkeit für die Meshes sowie für die (semi-)automatische Erstellung von Bearbeitungsskripten dar. Diese Skripte können von dem Commandline Tool verarbeitet werden.

Programmiersprache	C++
Lizenz	GPL
Unterstützte Import Formate	PLY, STL, OFF, OBJ, 3DS, COLLADA, PTX, X3D, VRML
Unterstützte Export Formate	PLY, STL, OFF, OBJ, 3DS, COLLADA, X3D, VRML, DXF
Vereinfachungsmethoden	Ja
Reparaturmethoden	Ja
Hierarchie Support	
LOD Generierung	Nein
Texturoperationen	Ja

Des Weiteren eignet sich Meshlab auch für die Verarbeitung von 3D Scans, genauer für die Ausrichtung mehrerer Teilscans und für die Reparatur der Oberfläche eines Scans.

## 9.3 JMeshLib

Dieses in C++ geschriebene Framework für die Bearbeitung von Meshes ist auf die Verwendung als Programmbibliothek ausgelegt, d.h. es gibt keine GUI. Es dient hauptsächlich zur Bearbeitung von 3D Scan Daten.

Programmiersprache	C++
Lizenz	GPL
Unterstützte Import Formate	OFF, PLY, STL, VER-TRI, VRML 1.0*, VRML 97*, OBJ, IV 2.1
Unterstützte Export Formate	OFF, PLY, STL, VER-TRI, VRML 1.0*, VRML 97*, OBJ, IV 2.1
Vereinfachungsmethoden	Ja
Reparaturmethoden	Nein
Hierarchie Support	
LOD Generierung	Nein
Texturoperationen	Nein

\*teilweise

## 9.4 PyMesh



Hierbei handelt es sich um ein Rapid Prototyping Framework bei dem der Schwerpunkt auf der Meshbearbeitung liegt. Bereitgestellt wird es als Programmbibliothek für Python. Es existiert keine Funktion für die Reduzierung von Meshgrößen, allerdings kann diese mit bereits bestehenden Funktionen implementiert werden.

<b>Programmiersprache</b>	Python
<b>Lizenz</b>	BSD 2
<b>Unterstützte Import Formate</b>	OBJ, OFF, PLY, STL, MESH, MSH, NODE, FACE, ELE
<b>Unterstützte Export Formate</b>	OFF, PLY, STL, VER-TRI, VRML 1.0*, VRML 97*, OBJ, IV 2.1
<b>Vereinfachungsmethoden</b>	Ja
<b>Reparaturmethoden</b>	Ja
<b>Hierarchie Support</b>	
<b>LOD Generierung</b>	Nein
<b>Texturoperationen</b>	Nein

## 9.5 OpenFlipper

Dieses Framework legt den Schwerpunkt auf das Rendern und die Bearbeitung von geometrischen Daten. OpenFlipper unterstützt neben Meshes auch NURBS Daten sowie Skelettbasierende Animationen. Ausgangsbasis für das Framework war ein Projekt der RWTH Aachen. Folgende technischen Details gelten für den zum Projekt gehörenden Sourcecode des Projekts:

<b>Programmiersprache</b>	C++
<b>Lizenz</b>	GPL
<b>Unterstützte Import Formate</b>	OBJ, OFF, STL, PLY, BVH, OM
<b>Unterstützte Export Formate</b>	OBJ, OFF, OM, PLY, VTK
<b>Vereinfachungsmethoden</b>	Ja
<b>Reparaturmethoden</b>	Nein
<b>Hierarchie Support</b>	
<b>LOD Generierung</b>	Nein
<b>Texturoperationen</b>	Nein

Allerdings verfügt OpenFlipper über ein QT Pluginsystem wodurch die Erweiterung des Frameworks über eine standardisierte Schnittstelle möglich ist.

## 9.6 Blender

Blender ist eine komplette 3D Grafiksuite mit welcher nicht nur Modellierung, sondern auch Texturierung und Animation möglich ist. Blender ist hauptsächlich auf die Verwendung der GUI ausgelegt, es ist allerdings möglich Skripte über die Commandline auszuführen. Blender ist wie OpenFlipper über ein Pluginsystem erweiterbar, die technischen Details beziehen sich also auf den „Out of the box“ Zustand.

<b>Programmiersprache</b>	C/C++ (Blender), Python (Skripts)
<b>Lizenz</b>	GPL
<b>Unterstützte Import Formate</b>	DAE, ABC, 3DS, FBX, BVH, PLY, OBJ, X3D, STL

Unterstützte Export Formate	DAE, ABC, 3DS, FBX, BVH, PLY, OBJ, X3D, STL
Vereinfachungsmethoden	Ja
Reparaturmethoden	Ja
Hierarchie Support	
LOD Generierung	Nein
Texturoperationen	Ja