Dokumentation Text2Scene
Gruppe 4.1
Tim Hau

Projekt starten

Das Projekt kann entweder über eine IDE oder über das Terminal gestartet werden. Hier nehme ich an, dass das Projekt auf Linux / Macos gestartet wird. Windows bleibt noch zu testen.
Wichtig ist hier, dass zuerst Eigen als dependency installiert wird. Anweisungen dazu im Readme.
Im Root folder müssen dafür folgende Befehle ausgeführt werden:

```
mkdir build
cd build
cmake ..
cmake --build .
```

Data directory

Wichtig ist, dass die Datensätze (Matterport, SUNC) in die richtigen Ordner eingefügt werden, da das Programm momentan nur auf dieser Ordnerstruktur aggieren kann.
Im Ordner /data/datasets sollte dafür schon die richtige Ordnerstruktur angelegt sein.
Nicht notwendigerweise erforderliche Ordner wie bspw. /debug werden vom Programm notfalls bei Bedarf selbst erzeugt.
Dabei variieren die Ordner zwischen den Datasets, da diese unterschiedlich aufgebaut sind

Für Matterport:

/cameras	--- exterior.cam
/config	--- hier kommt die .toml datei hin (wird notfalls erzeugt)
/debug	--- hier werden die .ply Objekte hin geschrieben
/house_segments	--- Analog zu den Dateien auf dem Server (.house, .ply etc.)
/metadata	--- category_mapping.tsv (war nicht auf dem server, von hier)
/xml	--- hier wird die .xml hingeschrieben (wird notfalls erzeugt)

Für SUNC:

/config	--- hier kommt die .toml datei hin (wird notfalls erzeugt)
/debug	--- hier werden die .ply Objekte hin geschrieben
/house	--- aufbau ist /house/<id>/house.json (vom server)
/object	--- Aufbau /object/<id>/<id>.obj
/room	--- Aufbau /room/<room_id>/<other_id>.obj
/xml	--- hier wird die .xml hingeschrieben (wird notfalls erzeugt)

verwendete Module

Das Projekt nutzt einige Open Source Bibliotheken, dabei sollten die meisten im git eingecheckt sein, da es sich um (kleinere) header Dateien handelt. Nur Eigen muss seperat installiert werden. Hier reicht es per git clone das Repo in den /include folder zu klonen und den Unterordner /eigen/Eigen nach /include zu verschieben.

Benutze Module:

Eigen	--- für alles was mit linearer Algebra zu tun hat
json.hpp	--- json parser
cpptoml.h	--- für .toml dateien
happly.h	--- für .ply dateien
tiny_obj_loader.h	--- für .obj dateien
tinyxml2	--- für .xml dateien

Programmablauf

Startet man das Programm wird ein kleiner Dialog auf dem Terminal ausgegeben, bei dem folgende Optionen zur verfügung stehen:

r	--- Einlesen der Datasets (beim ersten ausführen)
x	--- Export nach .xml
d	--- schreibt bounding boxen nach .ply
q	--- quit

Beim einlesen der Datasets wird außerdem jedes Objekt einzeln als .ply geschrieben. Dies dient hauptsächlich dem debuggen und kann sehr lange dauern. Ist das nicht erwünscht, kann das letzte Argument zur ::transform() methode von true auf false gesetzt werden. Ist dies der Fall werden die Objekte nicht einzeln geschrieben.

Verbesserungen

Für das SUNC set noch die Kamera und die label informationen. Momentan wird als label die id + random zahl benutzt.
Für Matterport wird immer das gesamte Haus auf einmal bearbeitet, da die Informationen die benötigt werden, um für jedes Objekt eine Transformation anzugeben, nur global vorliegen. Dies ist sehr rechenintensiv.
Die Berrechnung der Bounding box ist für Matterport3d noch recht aufwending und kann vereinfacht werden.
Debuggen ist relativ aufwendig und tests könnten vielleicht helfen.
Bounding box ansatz ist nicht gut. Besser wäre ein Ansatz mit Konvexer Hülle.
Außerdem kann der test, ob zwei Objekte tangent zueinander sind verbessert werden. Bspw. könnte eine Epsilon Umgebung gewählt werden, in der zwei Objekte als tangent angesehen werden.