OSM2World

3D rendering OpenStreetMap data

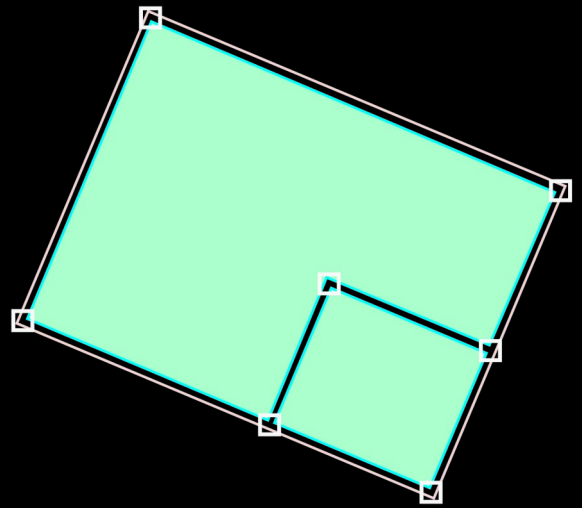
OpenStreetMap

Unterer Graben
Korbinianskreuzung
Schleimenbuckel
Alte Poststraße
Fischergasse
Erkerstraße
FS 44
Isarstraße

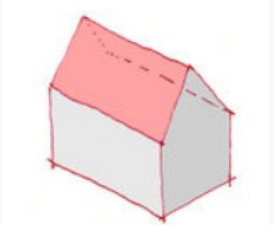
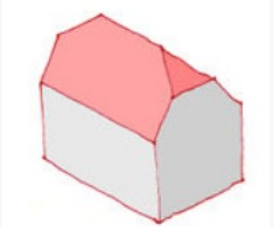
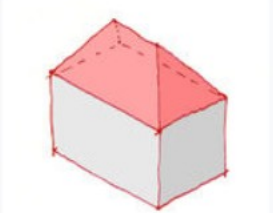
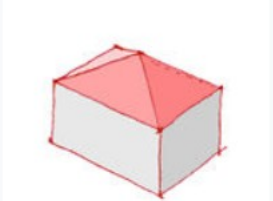
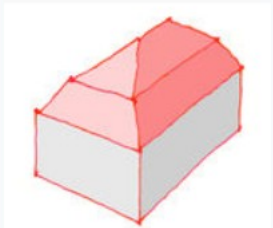
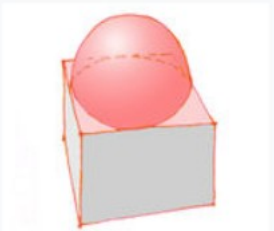
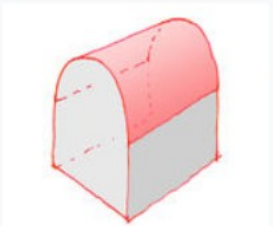
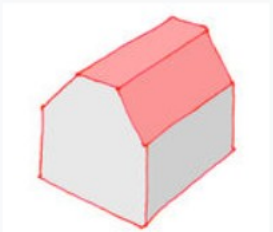
Input

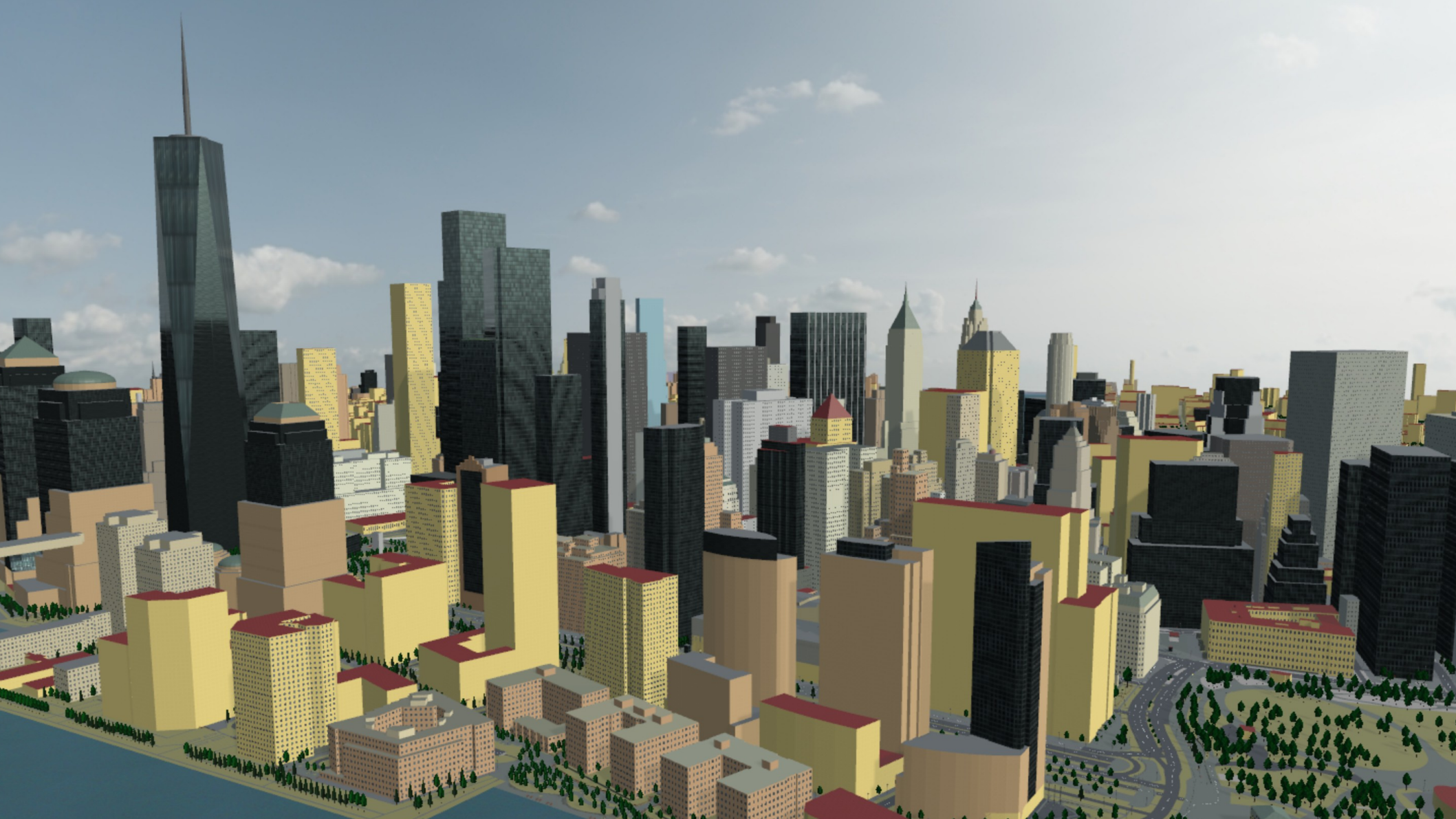
Simple 3D Buildings

- building, building:part
- height, building:levels, ...
- building:colour, roof:colour
- building:material, roof:material
- roof:shape, roof:direction, ...



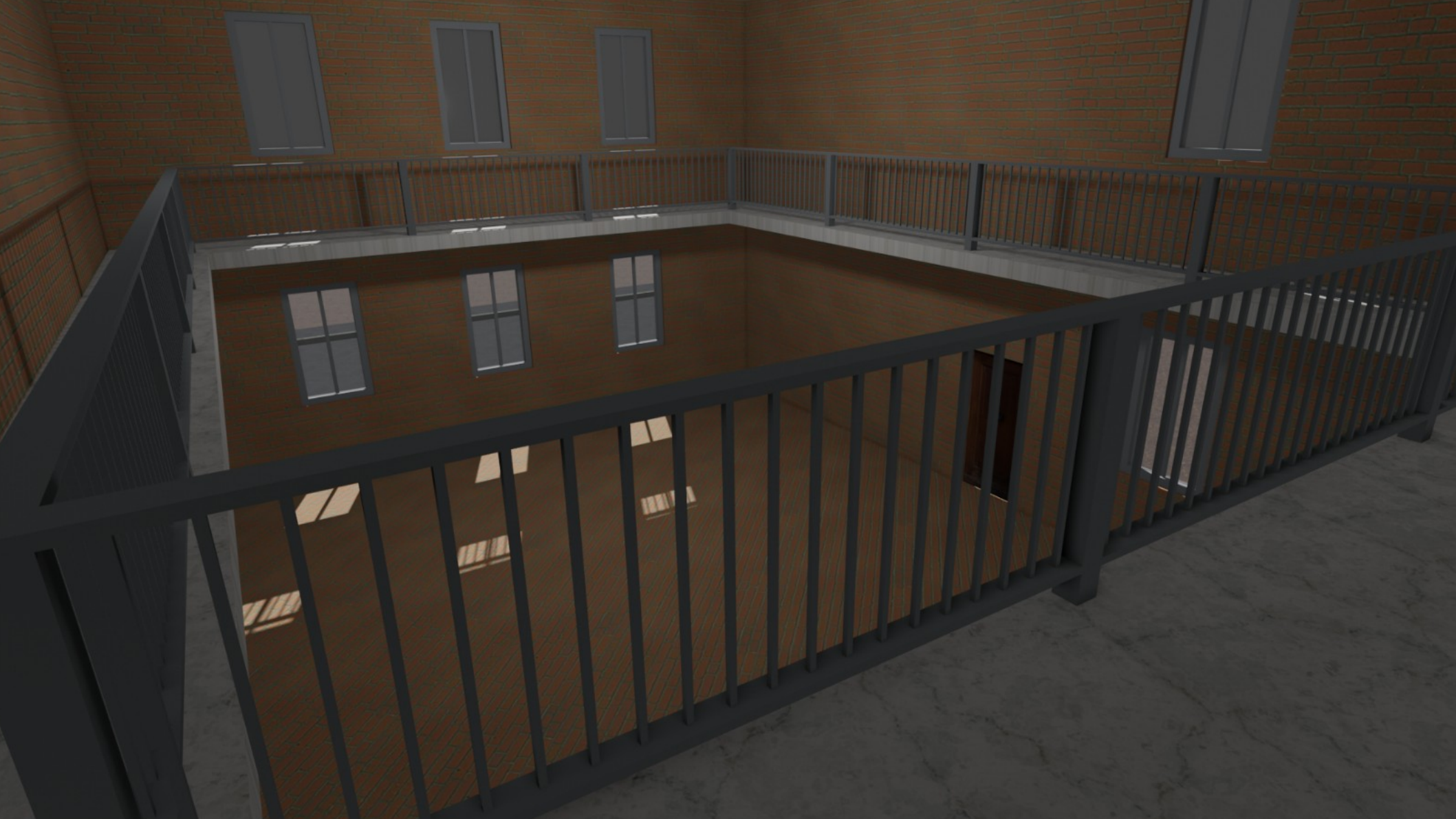
Simple 3D Buildings

					
flat	skillion	gabled	half-hipped	hipped	pyramidal
					
gambrel	mansard	dome	onion	round	saltbox



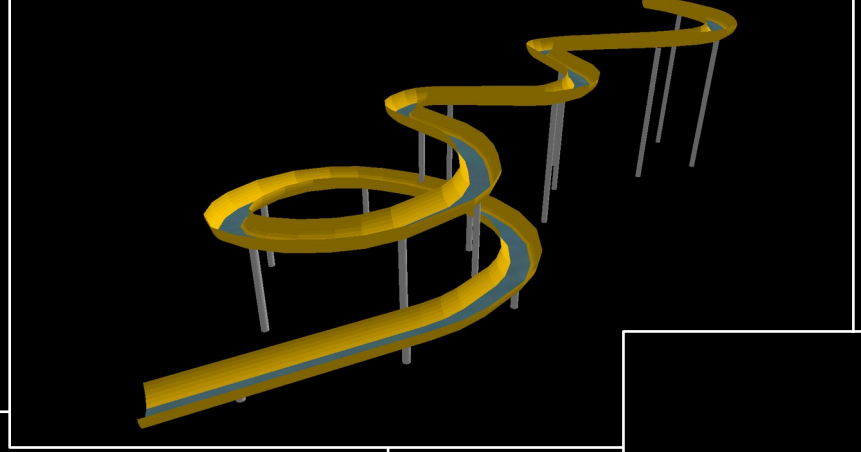
Input data

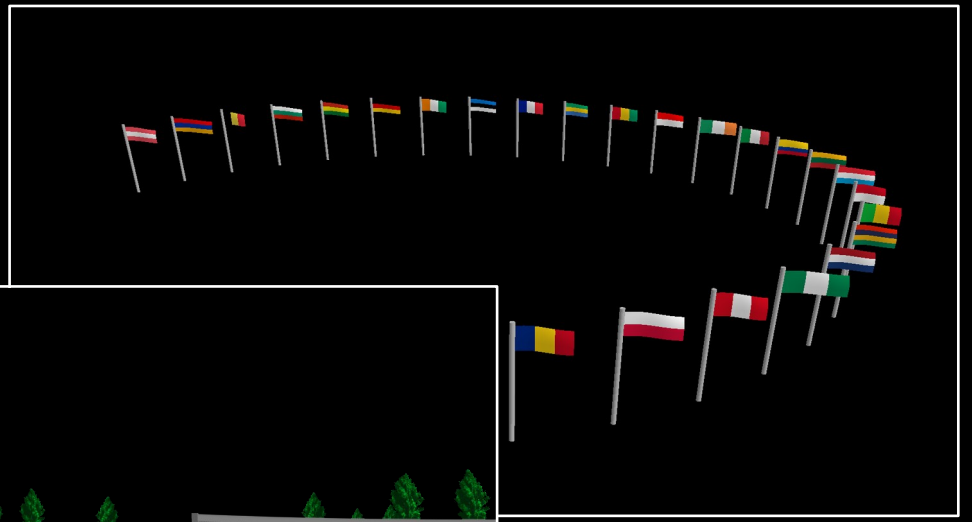
- Goal: Make the most of OSM data
- Simple 3D Buildings
- Simple Indoor Tagging

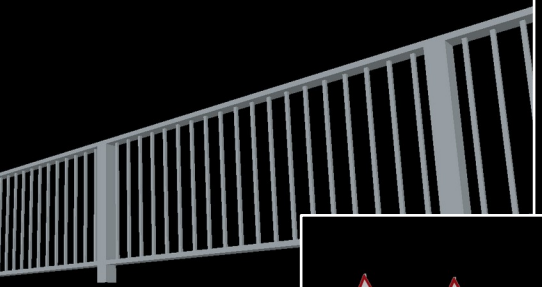


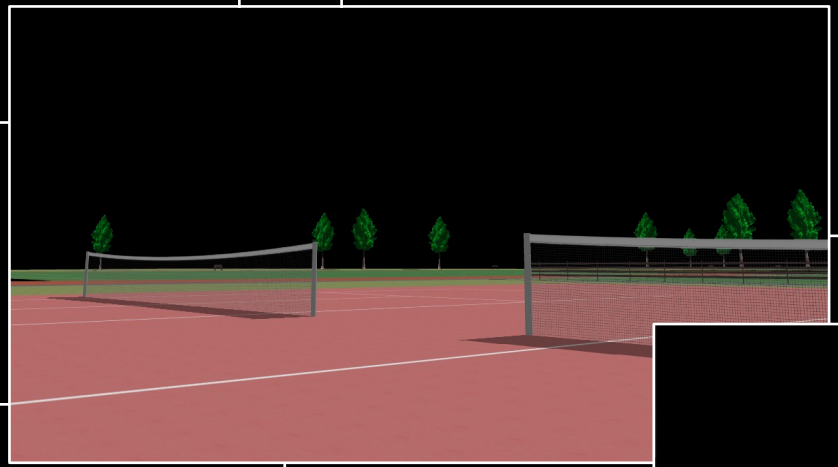
Input data

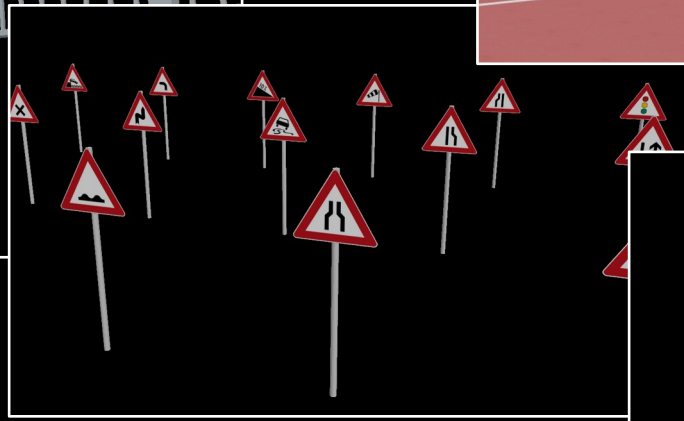
- Goal: Make the most of OSM data
- Simple 3D Buildings
- Simple Indoor Tagging
- Lanes and road markings
- 250+ other keys and tags

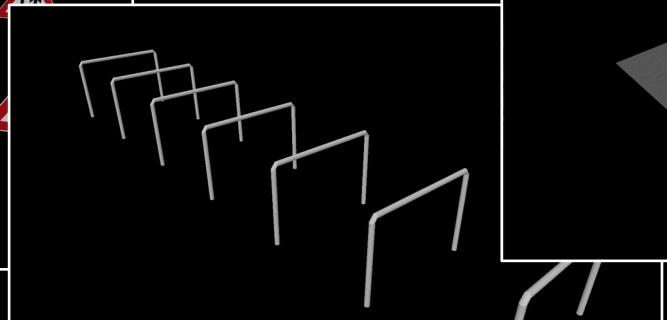


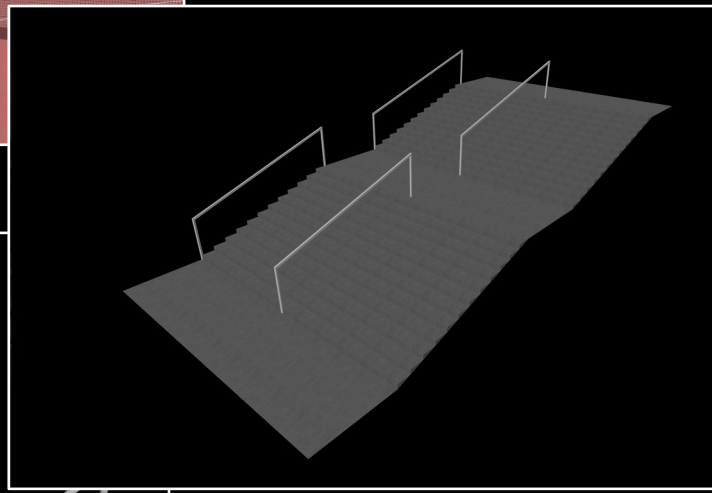












Input data sources

- Single files (OSM XML, JSON, PBF)
- Data tiles (PBF, e.g. with mapsplit)
- Overpass API
- MBTiles
- Geodesk

Output

Output formats

- 3D models from OSM data
- Exported as …
  - Model files (OBJ, glTF)
  - Images (PNG)
  - 3D Tiles
  - Interactive scenes with OpenGL/WebGL

glTF

- Open standard for 3D models, “JPEG of 3D”
- Most feature-rich OSM2World export format


File
Edit
Render
Window
Help
Layout
Modeling
Sculpting
UV Editing
Texture Paint
Shading
Animation
Rendering
Compositing
Scripting
Scene
View Layer
Global
Options
Object Mode
View
Select
Add
Object
User Perspective
(1) Collection
Scene Collection
Collection
Cube
Light
Sun
OSM2World scene
Camera
Background Scene
Active Clip
Units
Gravity
Keying Sets
Audio
Rigid Body World
Custom Properties
Playback
Keying
View
Marker
Start
End
1
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250

glTF

- Binary glTF (.glb), compressed output
- Single file or shared textures
- Post-processing, e.g.
  - Merging meshes per material
  - Emulating texture layers

3D Tiles

3D Tiles

- Optional: Creating .tileset.json
  → Compatibility with Cesium 3D Tiles
- WIP: Implicit Root Tiles

- Demo available on OSM2World Website
  Web frontend based on Babylon.js

Interfaces

Ways to use OSM2World

Ways to use OSM2World

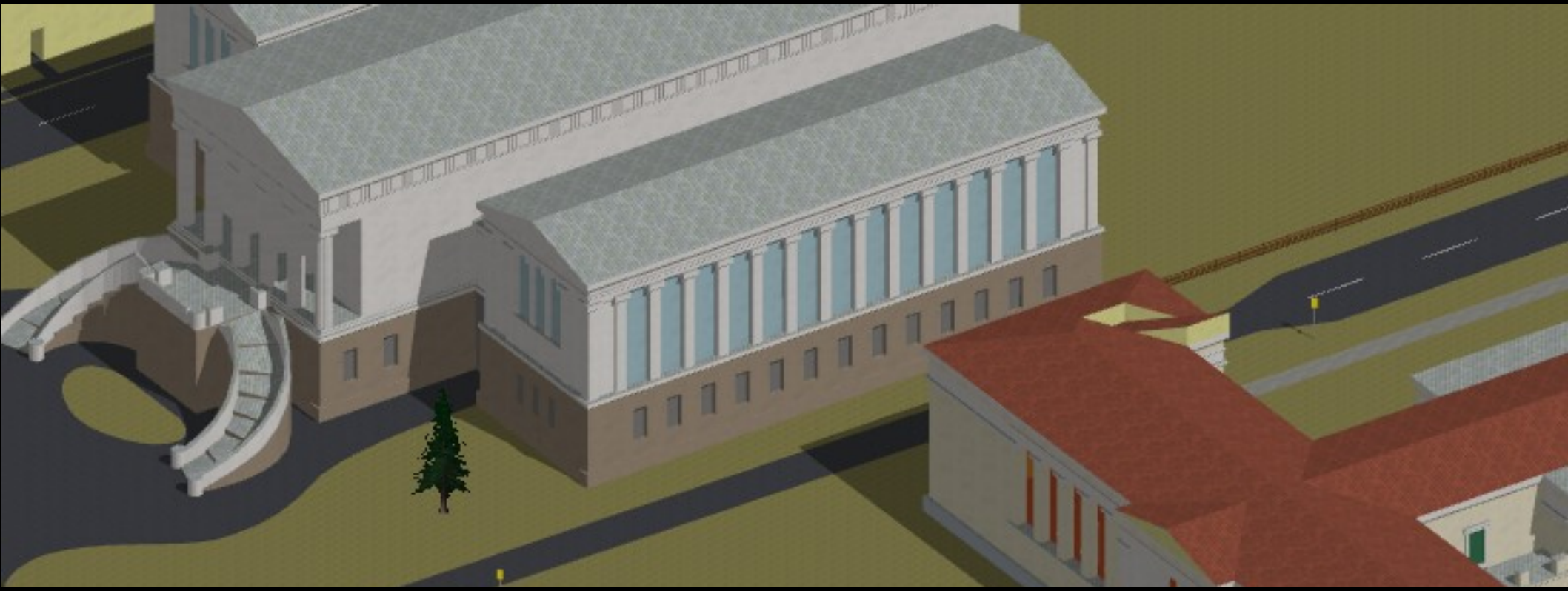
- 3D tile server
- Command-line interface
- Software library
- Desktop client
- OSM editor integrations (upcoming)

3D tile server

Command-line interface

```
Usage: osm2world [-hV] [COMMAND]
Creates 3D models from OpenStreetMap data
  -h, --help      Show this help message and exit.
  -V, --version   Print version information and exit.
Commands:
  help     Display help information about the specified command.
  convert  Convert OSM data to 3D models and output the result.
  gui      Start the graphical user interface.
  params   Run 'convert' with parameters from a text file or directory.
  server   Operate a tile server handling requests for 3D Tiles.
  tileset  Create 3D Tiles for some geographic area.
```

Command-line interface

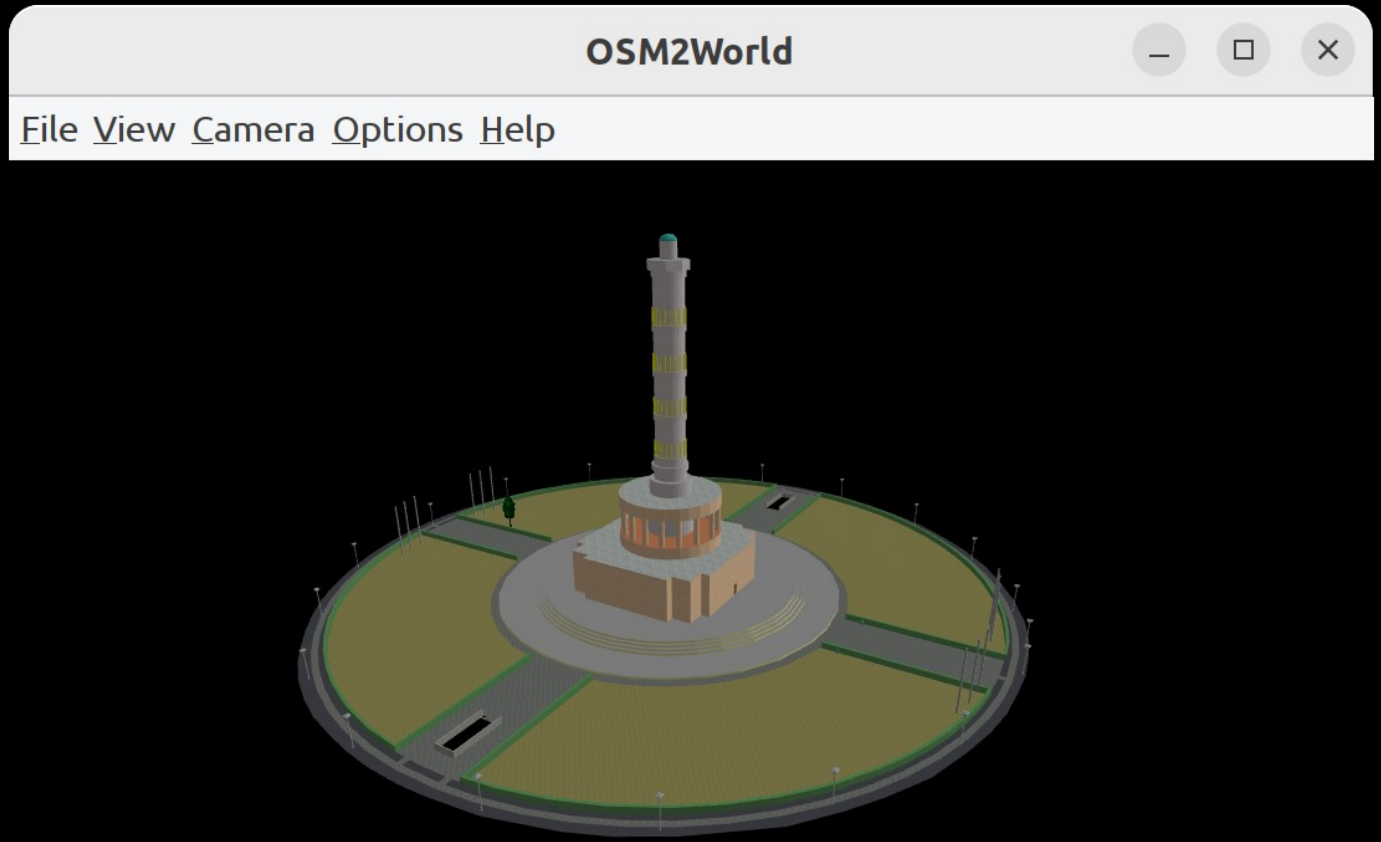


```
osm2world convert -i in.osm -o out.png
```

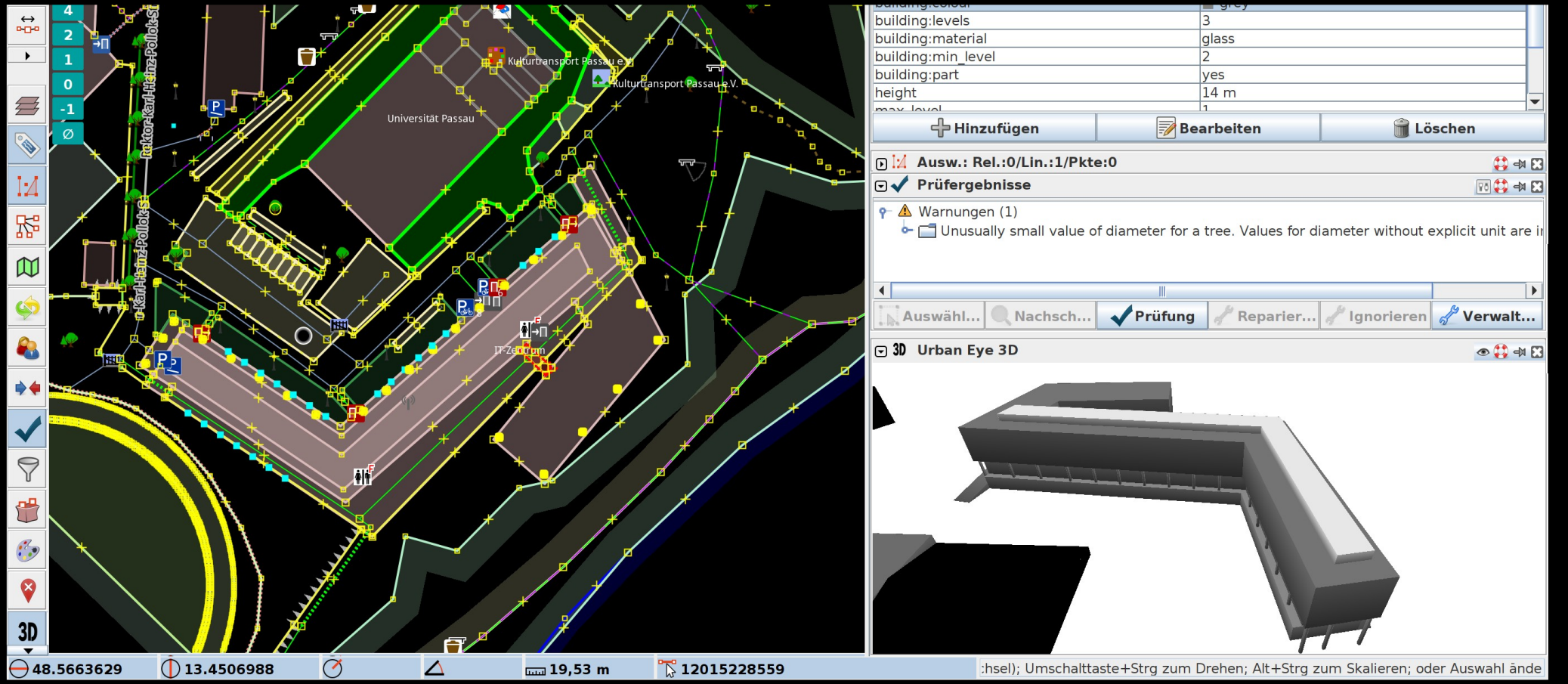
Library Interface

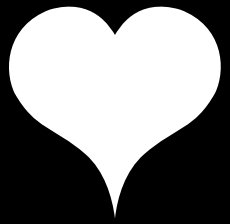
- Packages for Maven/Gradle:
  - osm2world-core
  - osm2world-opengl
  - osm2world-desktop
- https://osm2world.org/docs/library/
- Soon: JS library for client-side use

Desktop client



Editor integrations (upcoming)





Thanks to the Prototype Fund and the German Federal Ministry of Research, Technology and Space

osm2world.org/blog/2025/09/24/ptf-roadmap-2025-osm-3d-edit/

Feature Spotlights

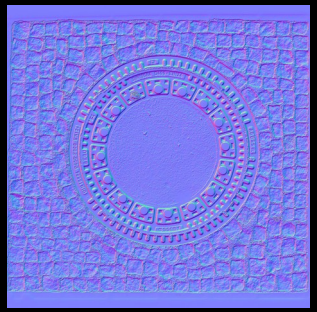

Feature Spotlight
Assets: Textures and Models

Textures

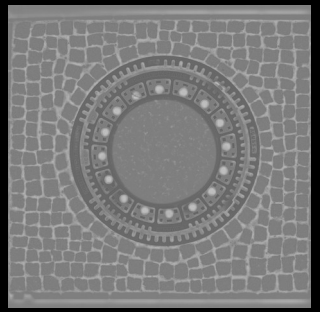
- Default style, customizable
- Raster images (JPEG, PNG)
- Vector images (SVG)
- Runtime generation (e.g. text)
- Physically Based Rendering

PBR Textures

Color

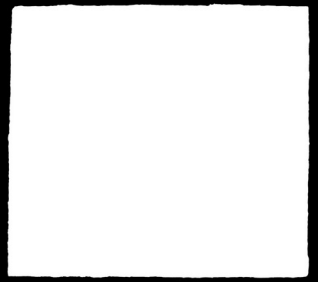


Normal

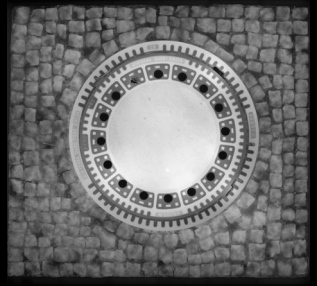


Roughness

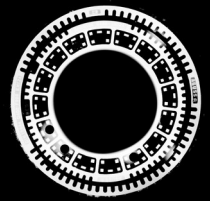
Occlusion



Opacity



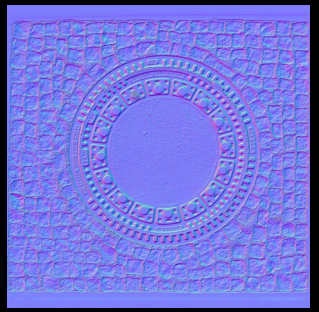
Displacement



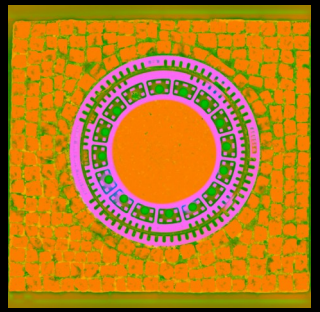
Metalness

PBR Textures

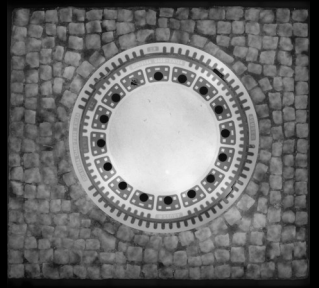
Color



Normal

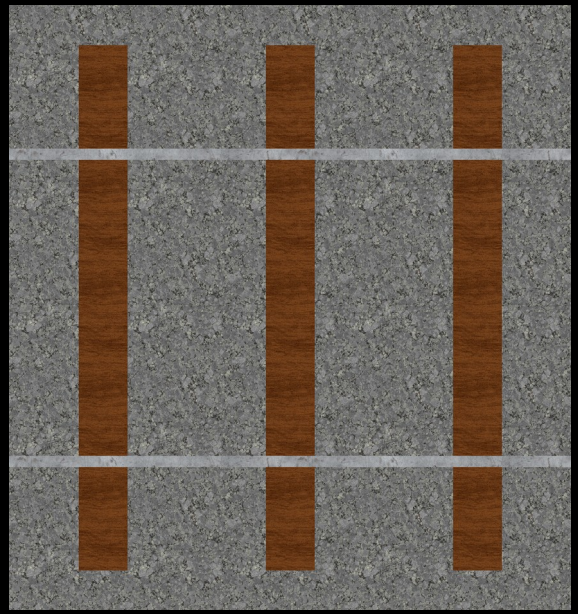


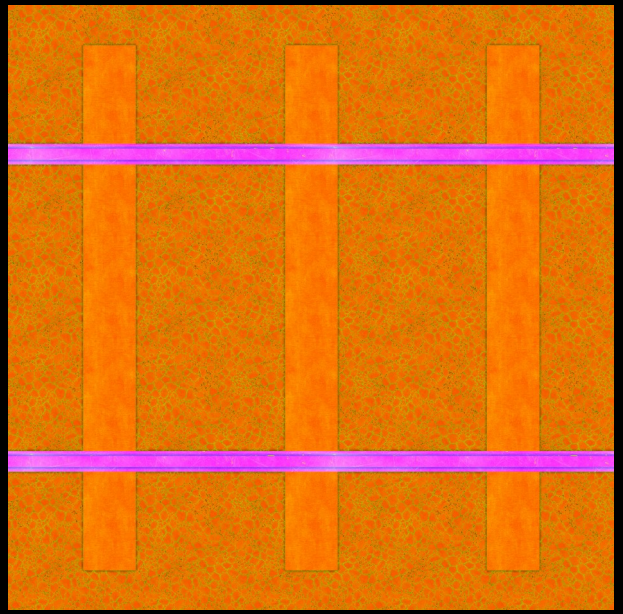
ORM

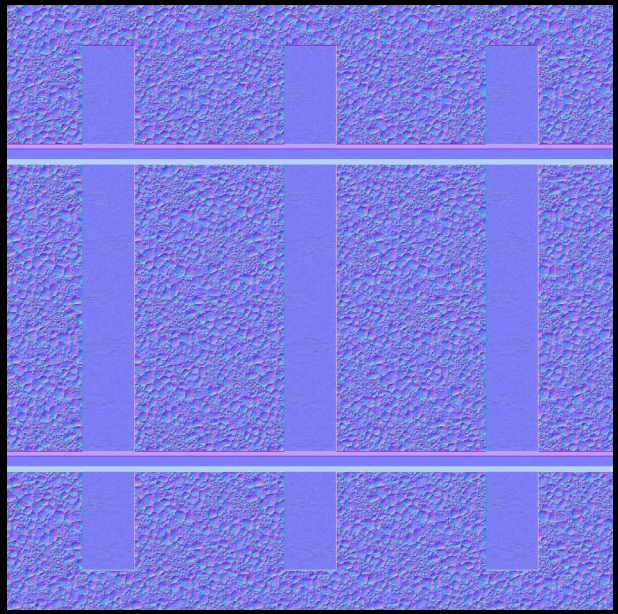


Displacement

Texture Camera

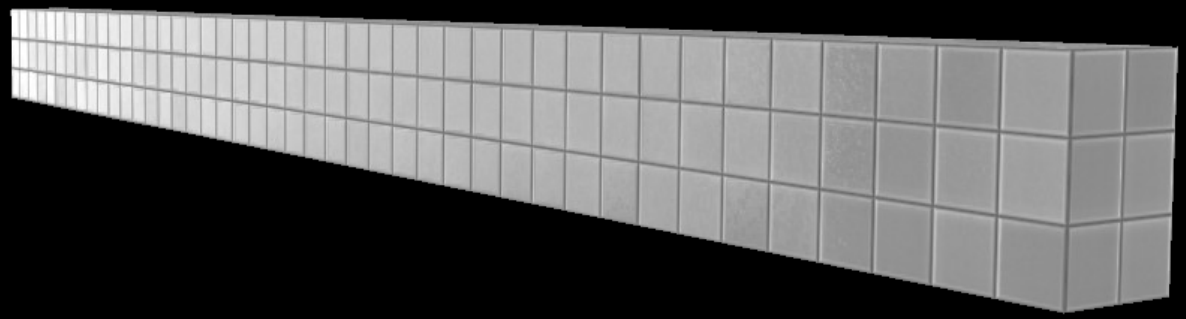






Texture Atlas

Texture Snapping

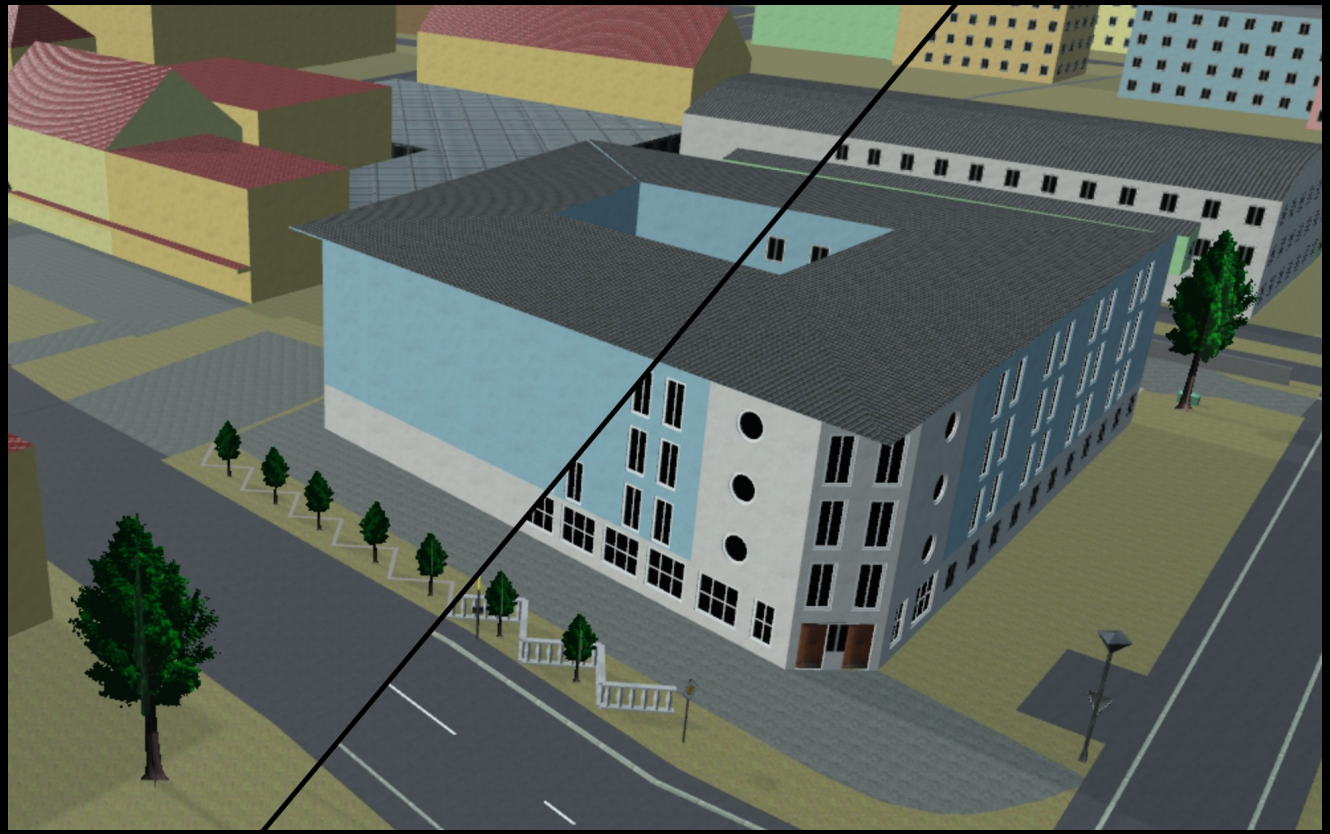


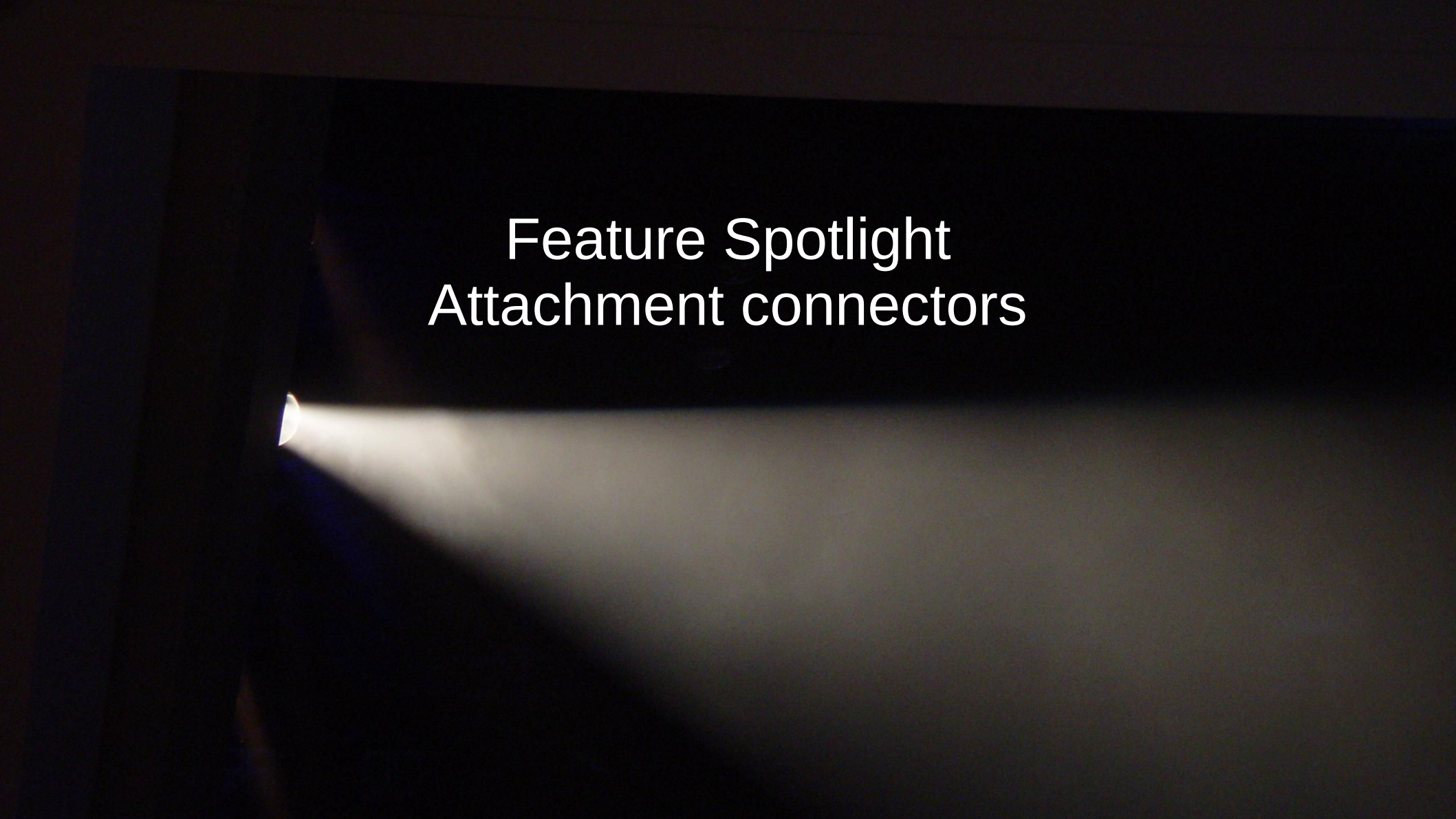
glTF Import

- 3D assets in addition to procedural models
- Random selection from alternative models
- Variable main color
- 3D Model Repository

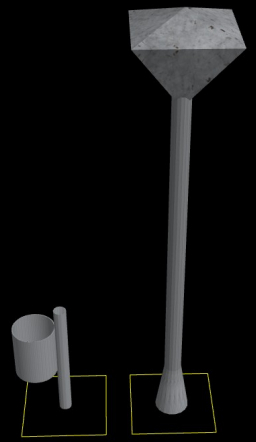

Feature Spotlight
Real-time performance and LOD

Level of Detail




Feature Spotlight
Attachment connectors

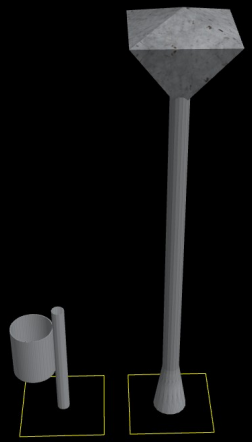
Attachment connectors



`amenity=waste_basket`

`highway=street_lamp`

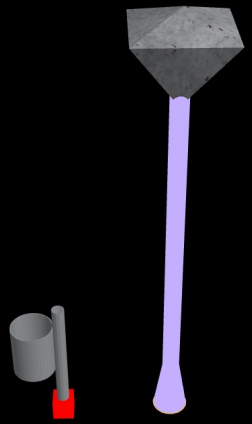
Attachment connectors



`support=street_lamp`
`amenity=waste_basket`

`highway=street_lamp`

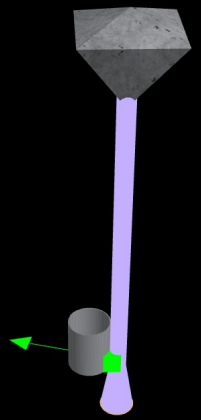
Attachment connectors



`support=street_lamp`
`amenity=waste_basket`

`highway=street_lamp`

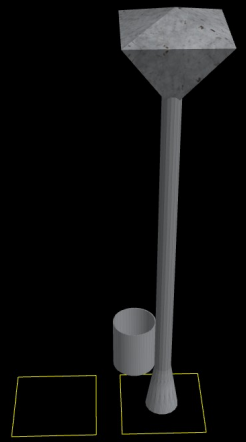
Attachment connectors



`support=street_lamp`
`amenity=waste_basket`

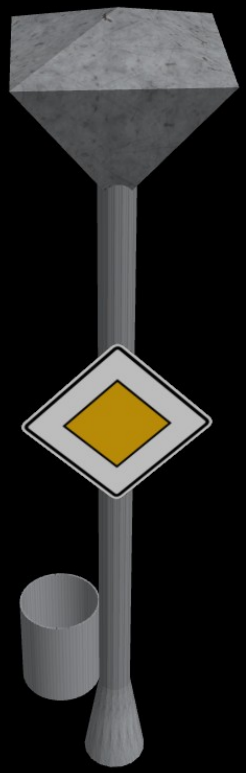
`highway=street_lamp`

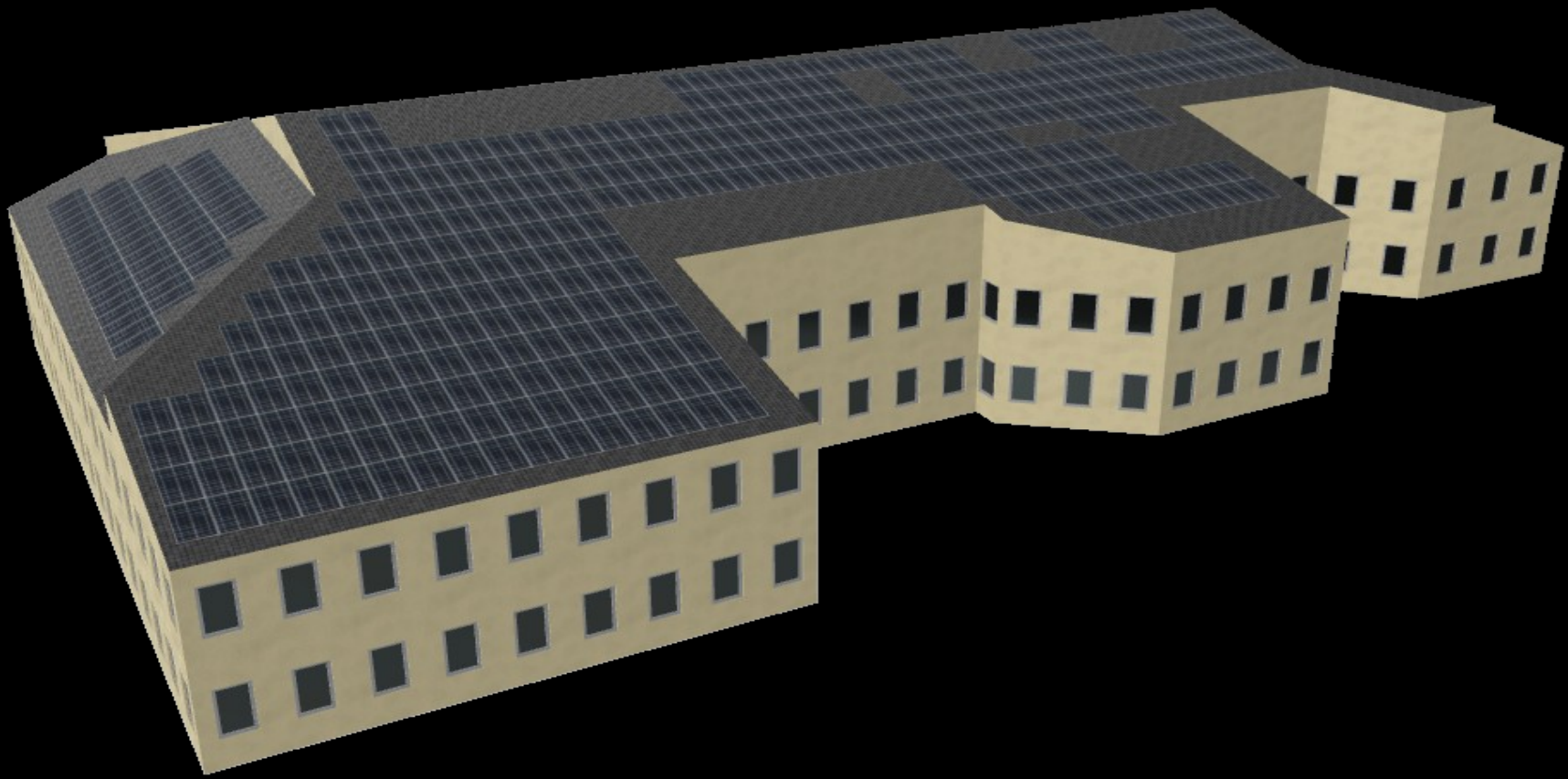
Attachment connectors



`support=street_lamp`
`amenity=waste_basket`

`highway=street_lamp`





Other attachment examples

- rooftop parking
- objects on outdoor and indoor walls
- objects on indoor floors
  - may attach to multiple levels, e.g. stairs
- objects on bridges

OSM2World

Get in touch!
https://tobias-knerr.de/contact/