Speaker notes

This talk discusses point clouds, and the py3dtiles tool and library for manipulating 3D Tiles and point clouds.

ÉRIC LEMOINE

Developer @ Oslandia
FOSS4G developer and enthusiast since 2007

eric.lemoine@oslandia.com
@elemoine
@erilem

Speaker notes

My name is Éric Lemoine. I work at Oslandia. And I've been working in the FOSS4G field since 2007.

OSLANDIA

Oslandia provides service on open-source software

- GIS
- 3D
- DATA

Speaker notes

Oslandia is an open-source company working on GIS, 3D and Data Science. QGIS, PostGIS and the iTowns 3D WebGL framework are examples of software components we are working on.

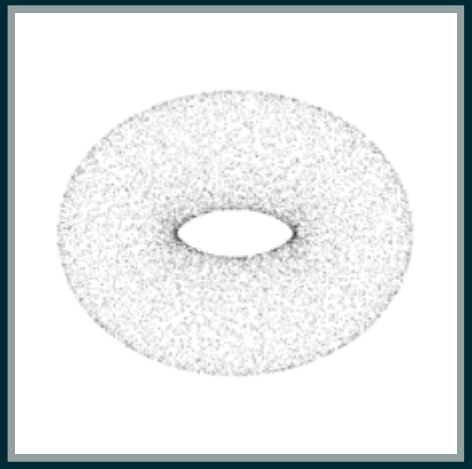
POINT CLOUDS!

Speaker notes

Let's talk about point clouds in general first!

POINT CLOUDS

« A point cloud is a set of data points in space. »



Speaker notes

A point cloud is just a set of data points in space. Nothing more. Point clouds provide a way to represent objects of our environment. A church and streets around it in the previous slide, and a donut here.

POINT CLOUDS

- Generally produced by 3D scanners (LiDAR)
- Can also be created using Photogrammetry

Speaker notes

What can produce point clouds? Point clouds are generally produced by 3D scanner. This is the LiDAR (Light Detection And Ranging) technology. Point clouds can also be produced using photogrammetry techniques (through homolog points).

LIDAR

Terrestrial, Airborne, Mobile, Unmanned




Speaker notes

There are several types of LiDAR acquisitions: Terrestrial (fixed tripods), Airborne (planes or helicopters), Mobile (Google Car like), and Unmanned (drones).

MANY APPLICATIONS!

- Create Digital Elevation Models (DEMs)
- Create 3D models
- Detect objects and obstacles
- etc.

Speaker notes

Point clouds have a wide range of applications. Examples include creating Digital Evelation Models, Digital Surface Models, 3D models, and detecting objects and obstacles. Autonomous cars use LiDAR! For the creation of 3D models, 3D surfaces are derived from point clouds.

3D TILES

An open specification for streaming massive heterogeneous 3D geospatial datasets

https://github.com/AnalyticalGraphicsInc/3d-tiles

Speaker notes

3D Tiles is an open specification created by AGI, the company behind Cesium. 3D Tiles defines a spatial data structure and a set of tile formats for streaming 3D geospatial content such as 3D buildings, BIM/CAD, Instanced Features, and Point Clouds.

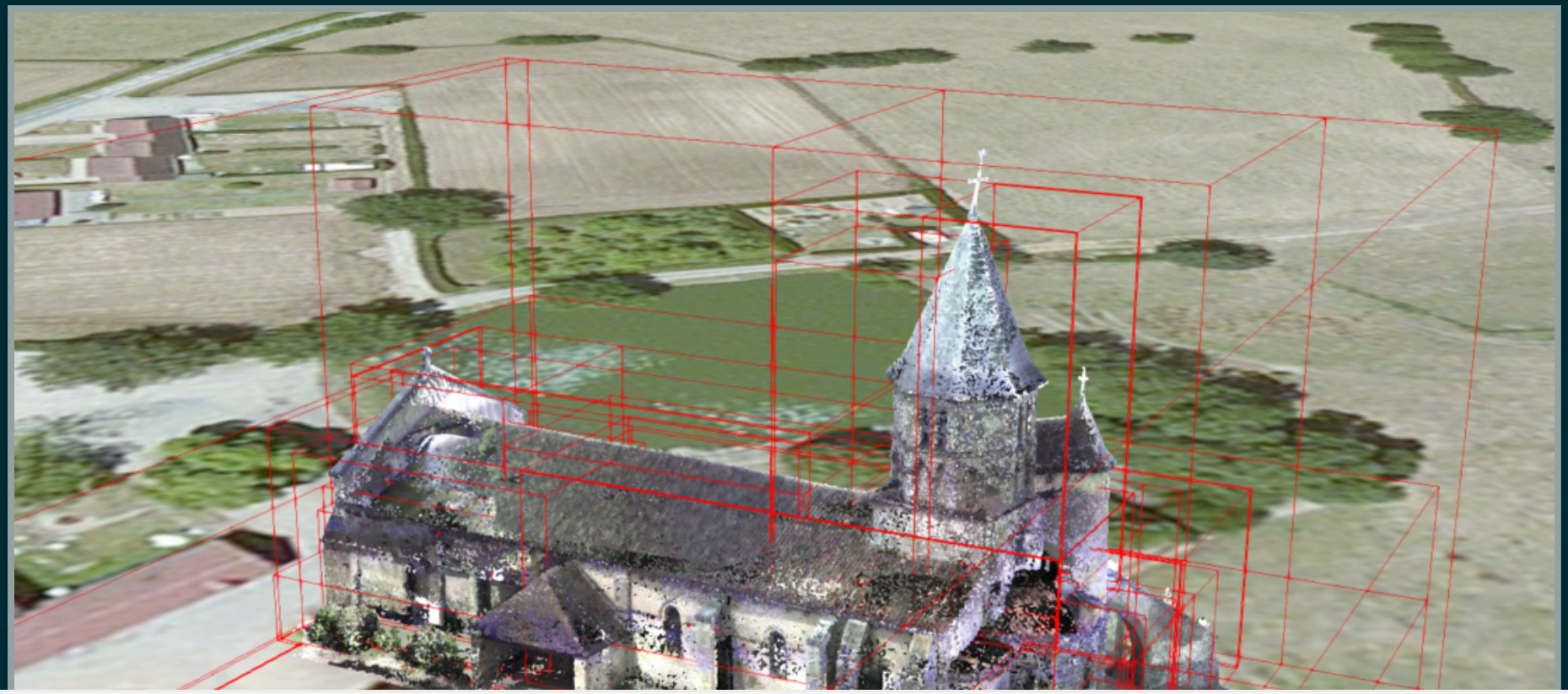
3D TILES

- tileset.json
  - Defines spatial layout for the tileset
  - Includes relatives paths to Tiles
  - May point to external tilesets
- Tiles
  - Contain the actual data
  - Binary formats

Speaker notes

A 3D Tiles tileset is composed of one or multiple tileset.json files defining the spatial layout for the tileset, and tiles that contain the actual data.

POINTCLOUD OCTREE



Speaker notes

This is an example of an octree for a point cloud. The 3D Tiles specification says nothing about the type of spatial layout to use. In this example an octree is used, but other spatial layouts such as quadtrees or kd-trees can be used as well. This is up to the person or tool generating the tile set.

TILE FORMATS

- Batched 3D Model (b3dm)
- Instanced 3D Model (i3dm)

Speaker notes

A Batched 3D Model tile contains heteregeneous 3D models, such as textured terrain and surfaces, 3D building exteriors and interiors. The glTF format is used for the model data.

An Instanced 3D Model tile contains 3D objects that are repeated in the scene with slight variations. Examples include trees, traffic lights, lamps etc. Here again the glTF format is used for the model data.

A Point Clouds tile contains points (of a point cloud). By default the point cloud data is not compressed, but the 3D Tiles specification defines an extension for compression the point cloud data with the Draco compression.

A Composite tile concatenates tiles of different formats into one tile.

PY3DTILES

A Python tool and library for manipulating 3D Tiles

https://github.com/Oslandia/py3dtiles

Speaker notes

py3dtiles provides both a CLI and API for manipulating 3D Tiles.

PY3DTILES CLI

`info`

read tile files and display a summary of their contents

`merge`

merge multiple tilesets into one

`convert`

create a pointcloud tileset from LAS files

PY3DTILES CONVERT

```
py3dtiles convert data/grandlyon/*.las \
    --srs_out 4978 --srs_in 3946 \
    --out lyon4978.3dtiles
```

Speaker notes

This is an example of a `convert` command to create a tileset from a set of LAS files. Here the points are reprojected from EPSG:3946 to EPSG:4978.

PY3DTILES CONVERT

Deal with infinite point clouds in finite

Speaker notes

We like to say that py3dtiles convert can deal with infinite point clouds in finites times.

Point clouds may be very big. Point clouds of multiple billions of points are the rule more than the exception. This means that, very often, a point cloud doesn't fit into memory.

For that reason py3dtiles convert reads and processes the data chunks by chunks. In this way py3dtiles convert may process point clouds of any size.

Also py3dtiles convert processes data chunks in parallel, using multiple processes, and trying to maximize the use of all the CPUs of the machine.

PY3DTILES PROCESSES

```
py3dtiles─┬─12*[py3dtiles─────5*[{py3dtiles}]]
          └─2*[{py3dtiles}]
```

- The main process
- 12 workers (by default)

Speaker notes

The main process spawns workers.

PY3DTILES TECHNOLOGIES

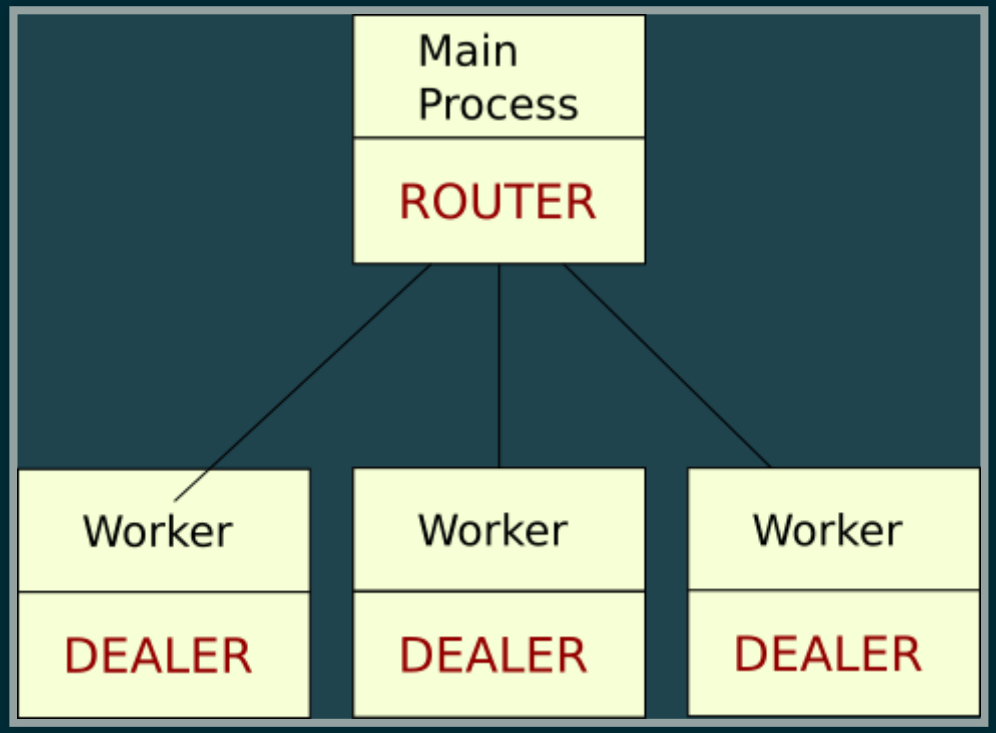
- Numpy
- ØMQ
- Numba

Speaker notes

py3dtiles uses top-notch Python technologies. py3dtiles uses Numpy for the manipulation of the point cloud data, ZeroMQ for balancing the load accross processes, and Numba to speed up the execution of hot code paths.

PY3DTILES ØMQ ARCHITECTURE

ØMQ's Load Balancing Pattern



(ROUTER Broker and DEALER Workers approach)

NUMBA

Example: processing of a ~2M-point point cloud

Speaker notes

Numba is an open source JIT compiler that translates a subset of Python and Numpy code into fast machine code.

Numba is quite easy to use – optimizating a function is just a decorator away! And the benefits may be huge!

We've used profiling (using Pyflame) to determine where py3dtiles was spending time, and then optimized some functions using Numba.

Numba is fast. Python is slow :(

ITOWNS / 3D TILES

Speaker notes

And now a video showing off an iTowns application displaying a point cloud of the city of Lyon in France. This point cloud has multiple billions of points. The points are colored by iTowns using images from a WMS.

PY3DTILES MAIN DEVELOPERS

- Jéméry Gaillard
- Ludovic Delauné
  ✉ ludovic.delaune@oslandia.com
- Pierre-Éric Pelloux-Prayer
  ✉ pierre-eric.peloux-prayer@oslandia.com

THANKS!

PERFORMANCE RESULTS

Entwine

```
time docker run -it -v $(pwd)/data:/opt/data \
    connormanning/entwine build \
    -i /opt/data/grandlyon -o /opt/data/entwine \
    -t 12 -r EPSG:3946 EPSG:4978 -f
```

```
real    79m24.884s
user    0m0.176s
sys     0m0.116s
```

```
time docker run -it -v $(pwd)/data:/opt/data \
    connormanning/entwine convert \
    -i /opt/data/entwine -o /opt/data/entwine-3dtiles
```

```
real    17m26.822s
user    0m0.064s
sys     0m0.012s
```

PERFORMANCE RESULTS

py3dtiles

```
time py3dtiles --verbose 1 convert data/grandlyon/*.las \
    --srs_out 4978 --srs_in 3946 --out lyon4978.3dtiles \
    --rgb 1 --overwrite 1
```

```
real    423m6.207s
user    3397m20.820s
sys     107m20.284s
```