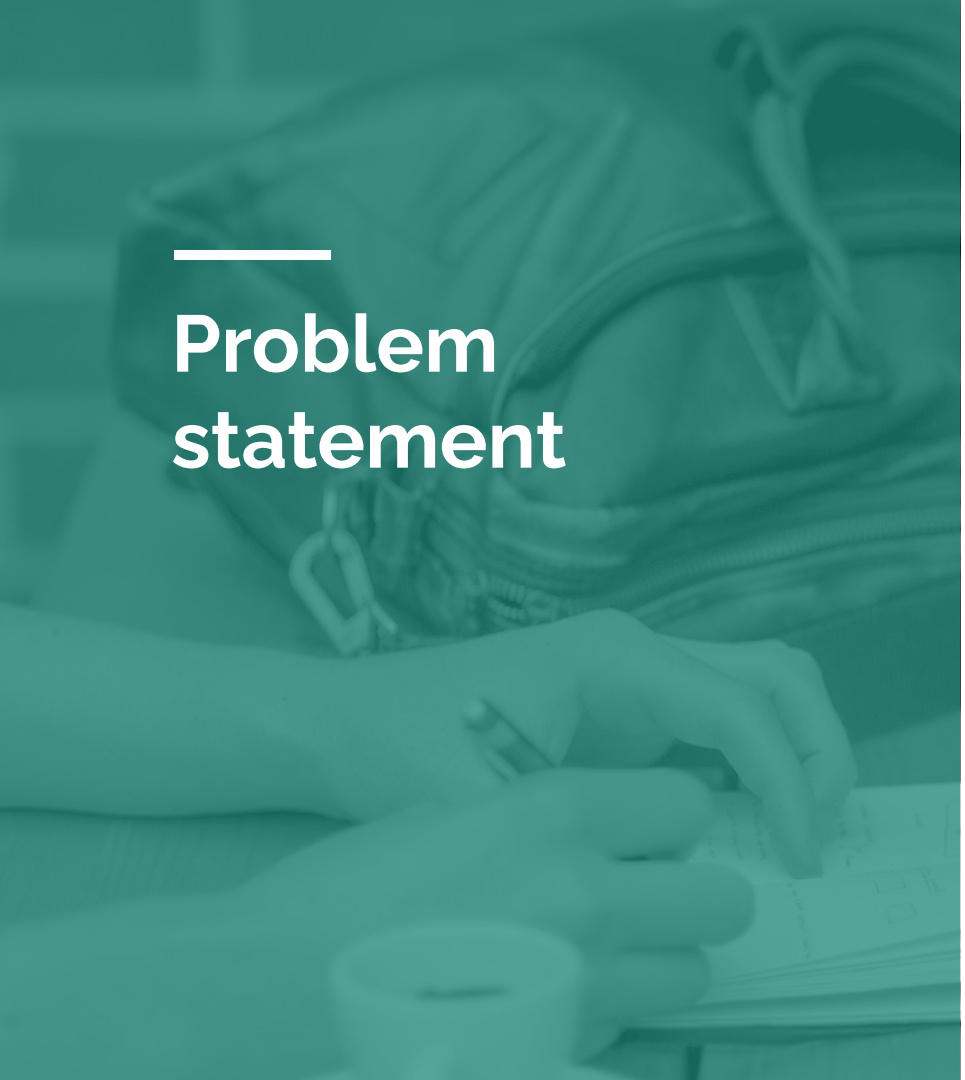
Mapillary

Keep maps detailed and up to date

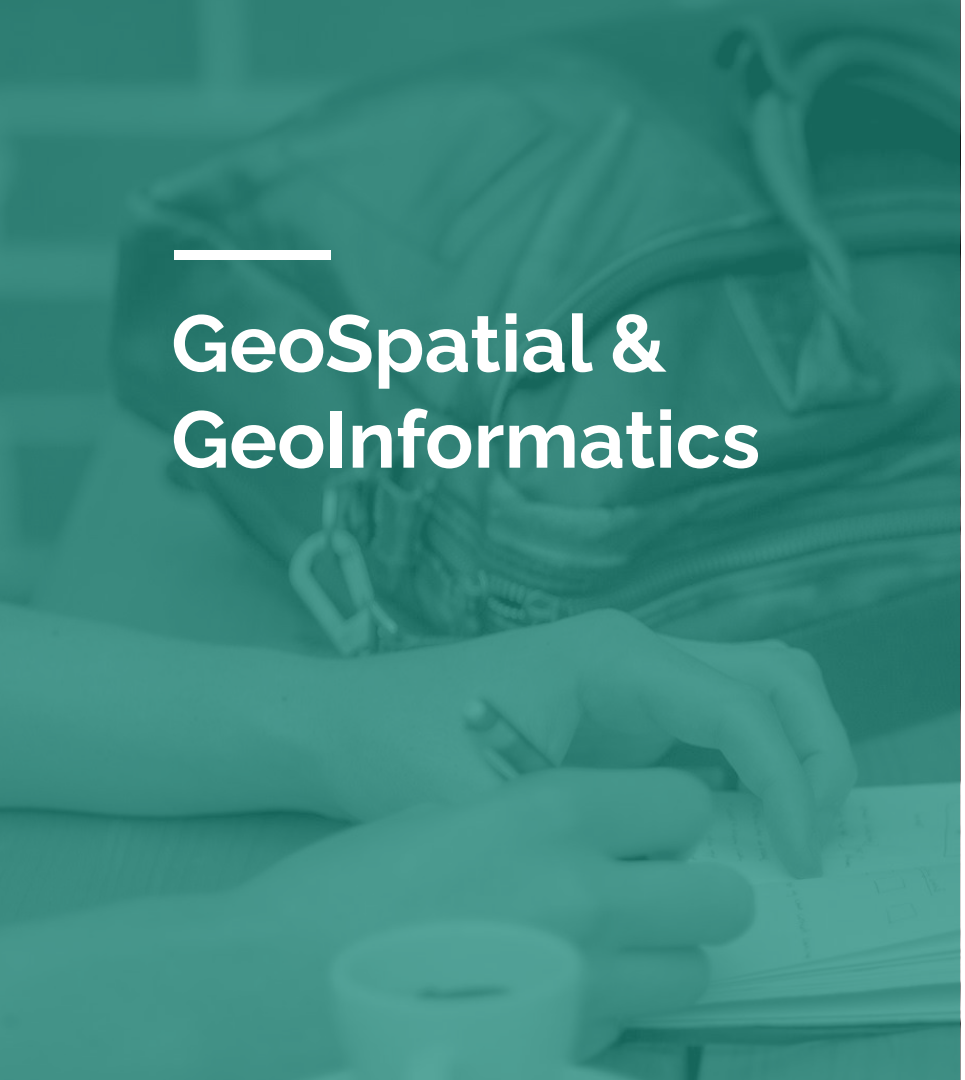
See what places look like through street-level imagery and add more details to maps.

Mapillary



Problem statement

Keeping maps easy to update and fresh for use



GeoSpatial & GeoInformatics

- MapBox
- Esri
- ArcGis

Solution Proposal

Open Source technologies

Mapillary provides,

1. **Map updation** via uploading images from **Desktops/Mobile/GoPro**
2. **Tools** for helping upload material
3. More than **75%+** of data on Mapillary is contributed by users!
4. **Structure From Motion** (SFM) - **2D to 3D** pixel forecasting for building 3D structures
5. Pipeline for **Computer Vision**
6. **MapillaryJS** - Interactive, extendable street imagery visualization platform
7. **ArcGIS** support
8. **OpenSFM** library for Structure From Motion

So why use Mapillary?

Why not stick to Google maps?

Pros: It’s Google
Cons: It’s only Google

Mapillary's Python SDK

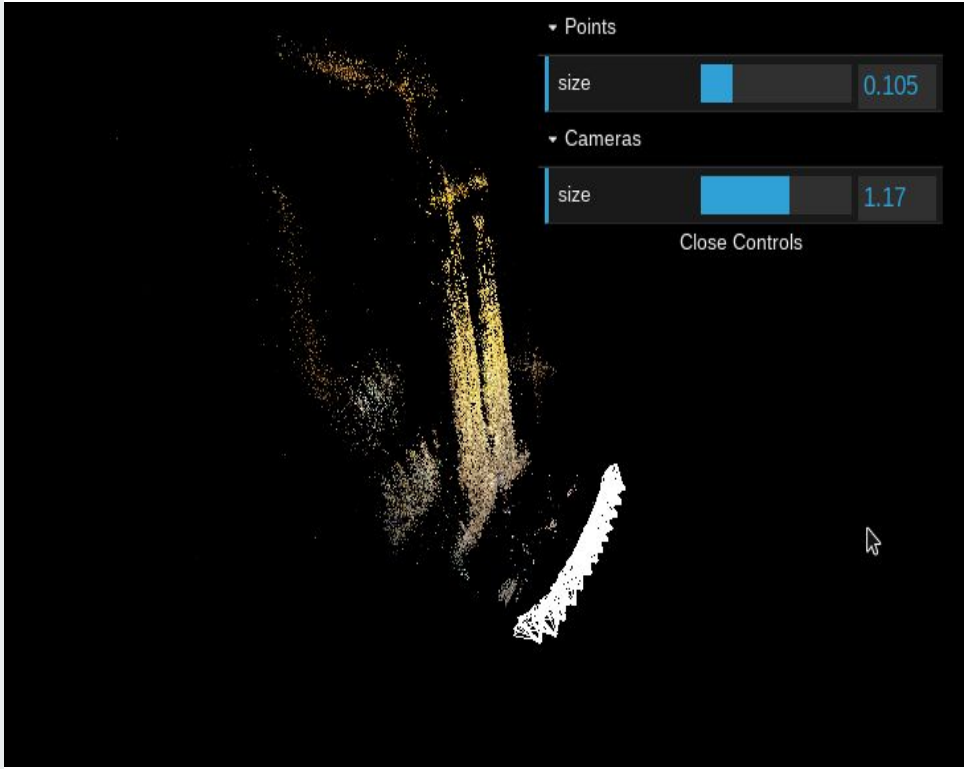
An SDK for interacting with the most recent Mapillary API v4 release, which introduced a lot of changes. While also catering to future possible API release 1-2 years down the road,

1. From scratch implementation, we only had a **Product Requirement Document** (**PRD**) to start with
2. **Forward Compatibility**, implementing **DRY**. Focusing on **maintainability**, **readability**, **modularity**, **structure**, and **best practices**
3. So far, **65 issues** (**44 closed**), **43 PRs** (**39 merged**), **7.9K+ LoCs**, **305 commits**
4. Preparing for a **release**, **documentation**, and writing **unit tests**
5. Already incoming traffic with **1,600+ views**, and **150+ unique visitors**
6. Thinking about being **Open Source** friendly as possible. **Facebook** guidelines follow up
7. Currently under facebookincubator

Open Sourcing 3D City Reconstruction

01

One of the downsides of using simple devices for mapping the world is that the GPS accuracy is not always great, especially in cities with tall buildings. Since the start we have always wanted to correct this using image matching and we are now making progress in that area.



Getting Environment Information

02

Develop your algorithms with diverse data from all over the world, including different weather, season, time of day, camera, and viewpoint conditions.

API Documentation “At A Glance”

03

All of the API Documentation for APIv4 is available here

- Root endpoint for metadata - https://graph.mapillary.com
- Root endpoint for vector tiles - **https://tiles.mapillary.com**
  - Coverage tiles - **https://tiles.mapillary.com/maps/vtp/mly1_public/2/{z}/{x}/{y}**
  - Computed coverage tiles - **https://tiles.mapillary.com/maps/vtp/mly1_computed_public/2/{z}/{x}/{y}**
  - Map features (points) tiles - **https://tiles.mapillary.com/maps/vtp/mly_map_feature_point/2/{z}/{x}/{y}**
  - Map features (traffic signs) tiles - **https://tiles.mapillary.com/maps/vtp/mly_map_feature_traffic_sign/2/{z}/{x}/{y}**

Team Lead

04

In the team

Christopher Beddow

Team Lead // Map Building At Facebook Reality Labs

Helping and guiding the direction of the project, prioritizing, answering every single question we have & introducing us to many on Mapillary's team!

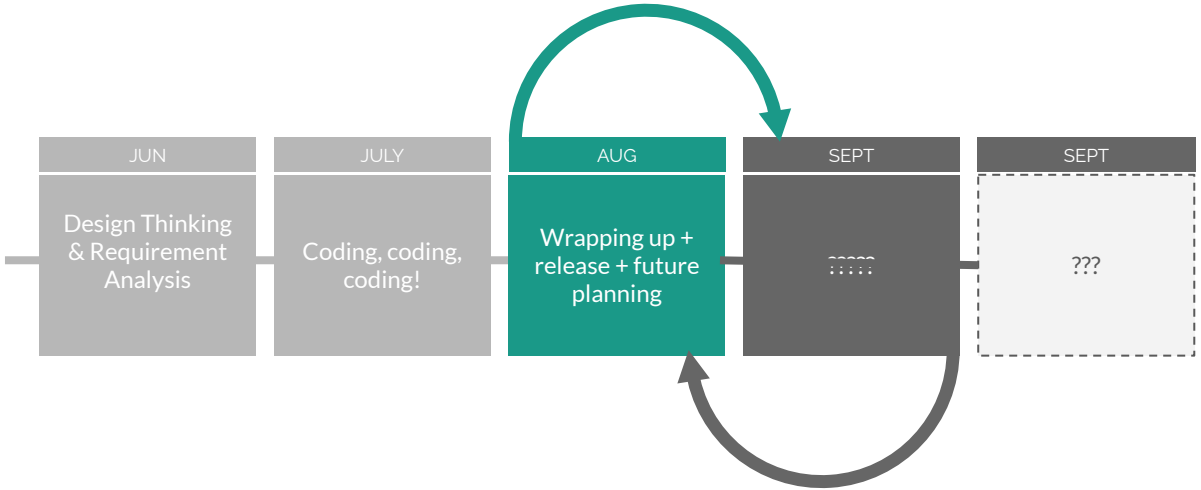
Why it’s better than existing solutions

Open Source!

What next?

- ➔ After the documentation and PyPI release? We're not sure
- ➔ Finishing a few left over issues on GitHub

Timeline


JUN
Design Thinking & Requirement Analysis
JULY
Coding, coding, coding!
AUG
Wrapping up + release + future planning
SEPT
.....
SEPT
???

What we need help with

- ➔ A few good issues, more about project maintenance
- ➔ More coming in the future (to infinity and beyond)

Questions?

References

Mapillary Python SDK

API Documentation

GitHub - Mapillary