8TH INTERNATIONAL ISPRS WORKSHOP

LOWCOST 3D - SENSORS, ALGORITHMS, APPLICATIONS

LIÈGE université GeoScITY

Automatic Detection and 3D Modeling of City Furniture Objects using LiDAR and Imagery Mobile Mapping Data

Doi Hiba[1], Anass Yarroudh[2], Imane Jeddoub[2], Rafika Hajji[1], Roland Billen[2]

[1] College of Geomatic Sciences and Surveying Engineering, Hassan II IAV, Rabat 10101, Morocco
[2] GeoScITY, Geomatics Unit, UR SPHERES, University of Liège, Belgium

Thursday, 12/12/2024: 14:20 – 14:40

Università degli Studi di Brescia, Brescia, Italy

About the project

- ❑ Final year master project at the College of Geomatic Sciences and Surveying Engineering, Hassan II Institute of Agronomy and Veterinary Medicine, in collaboration with the GeoScITY lab.

Erasmus+

International Credit Mobility

Higher Education Student and Staff Mobility between programme and partner countries

LIÈGE université
GeoScITY

معهد الحسن الثاني للزراعة و البيطرة

INSTITUT AGRONOMIQUE ET VÉTÉRINAIRE HASSAN II

Plan

- Background
- Motivation & Objectives
- Methodology
- Results and discussion

Background

Importance of comprehensive 3D City Models

- Role in urban planning, infrastructure management, environmental analysis.
- Technological backbone of urban digital twins.

Motivations

Although **city furniture** objects like traffic lights, traffic signs, poles and bus stations play crucial role in the urban tissue, current research does not provide a complete method to automatically detect, localize and model these objects in accordance with 3D city models standards.

Motivation 1

Enrich existing 3D city models by integrating accurate and detailed representations of city furniture

Motivation 2

Address a significant gap in the current state of the art by providing a complete and integrated workflow that includes the detection, localization and 3D modeling of city furniture

Objectives

Objective 1

Automatically detect various types of city furniture in images and LiDAR point clouds using deep learning techniques

Objective 2

City furniture positioning from images and LiDAR data

Objective 3

Retrieve features and characteristics of the detected urban furniture objects

Objective 4

Automated/parametric 3D modeling of city furniture using the extracted localization and features

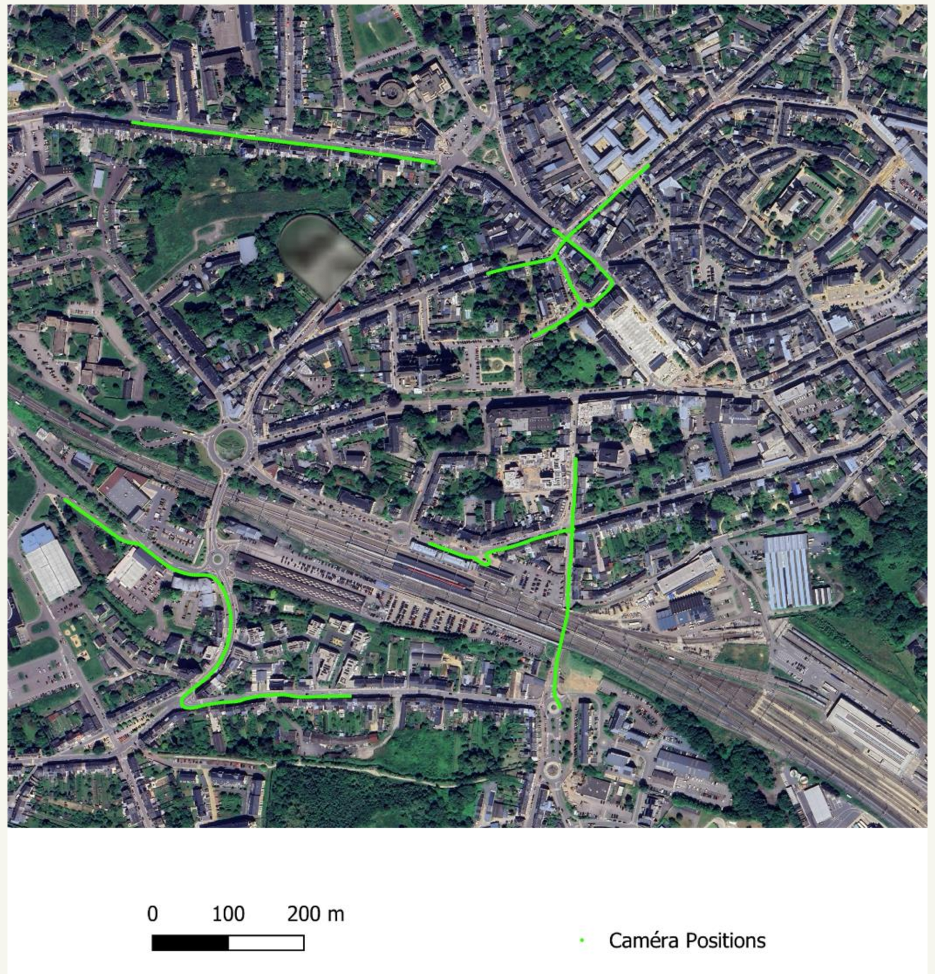
LIÈGE université
GeoScITY

Mobile Mapping Data

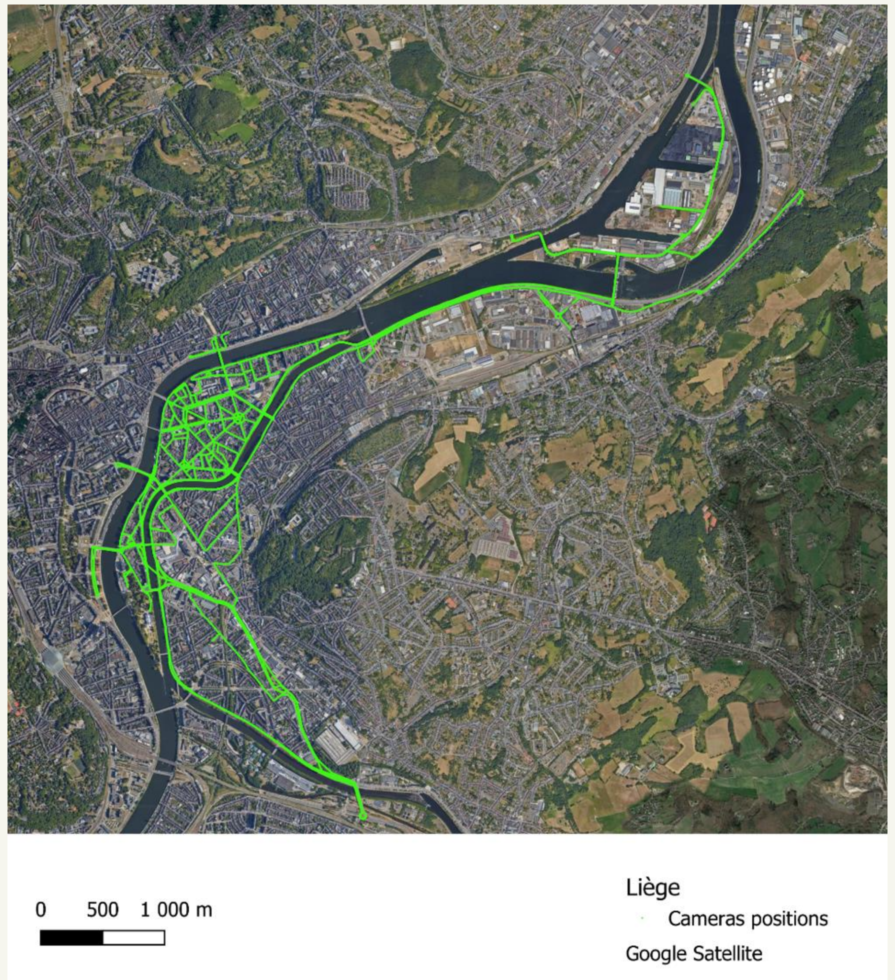
- Context
- Motivations
- Objectives
- **Imagery-based Approach**
- Camera-LiDAR Approach
- Modeling Approach
- Discussion
- Conclusion and Perspectives

Study area

Arlon City Belgium



Liège City Belgium



Dataset	Arlon	Liège
Provider	GlobeZenit	DrivenBy
MMS equipement	LiDAR, Panoramic camera, GNSS/IMU	Panoramic camera, GNSS/IMU
Data	360° images + Camera pose + LiDAR point clouds	360° images + Camera pose

LIÈGE université
GeoScITY

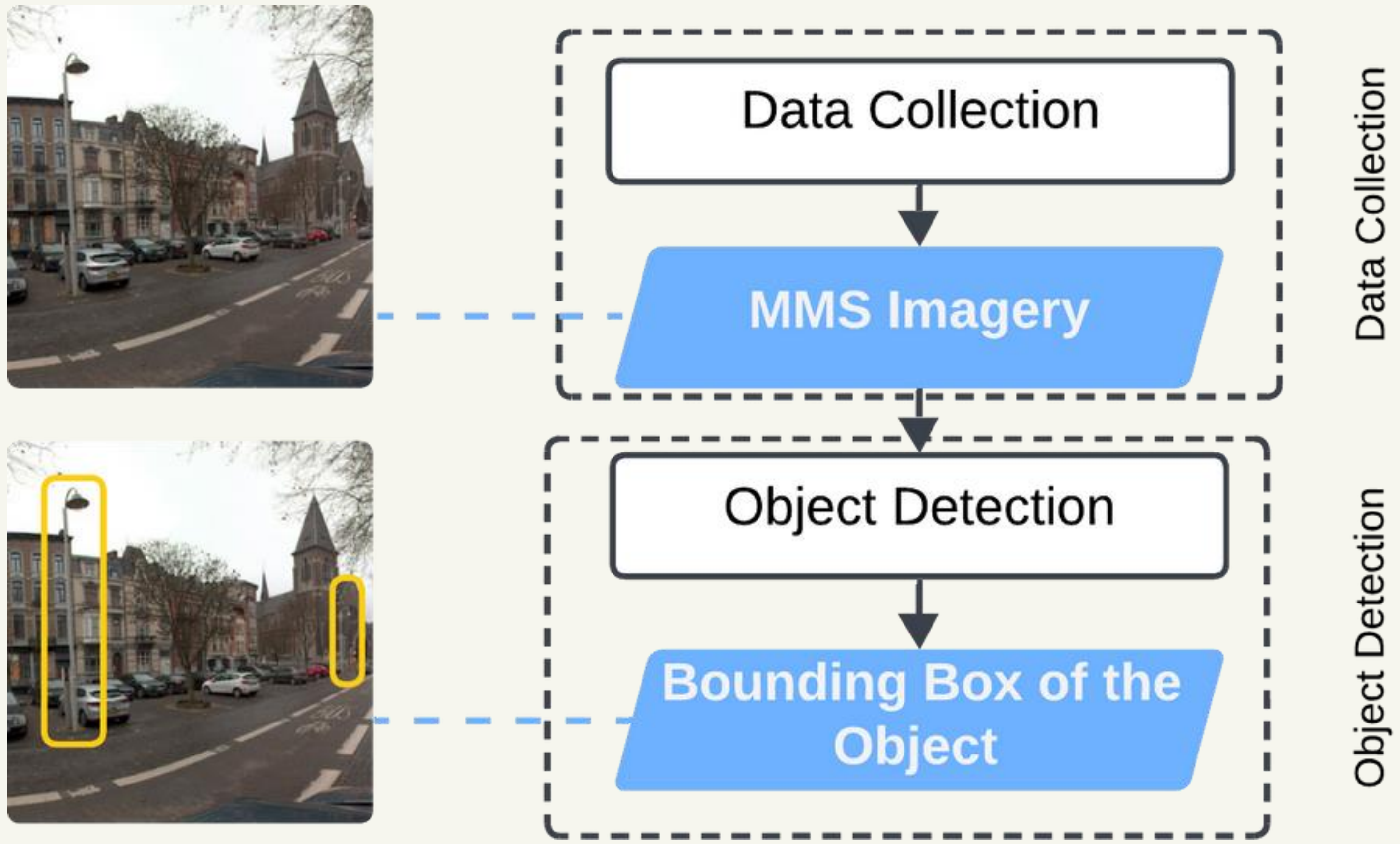
Imagery-Based Approach

Object detection

Prepare Dataset

Grounding dino for reducing manual labeling

Direct Object Detection		Cascaded Detection and Classification	
Traffic light	1 classes	Traffic sign	60 classes
Lamppost	3 classes		
Bus Stop	1 classes		



Training The models

Training multiple **YOLOv8** models

Object detection

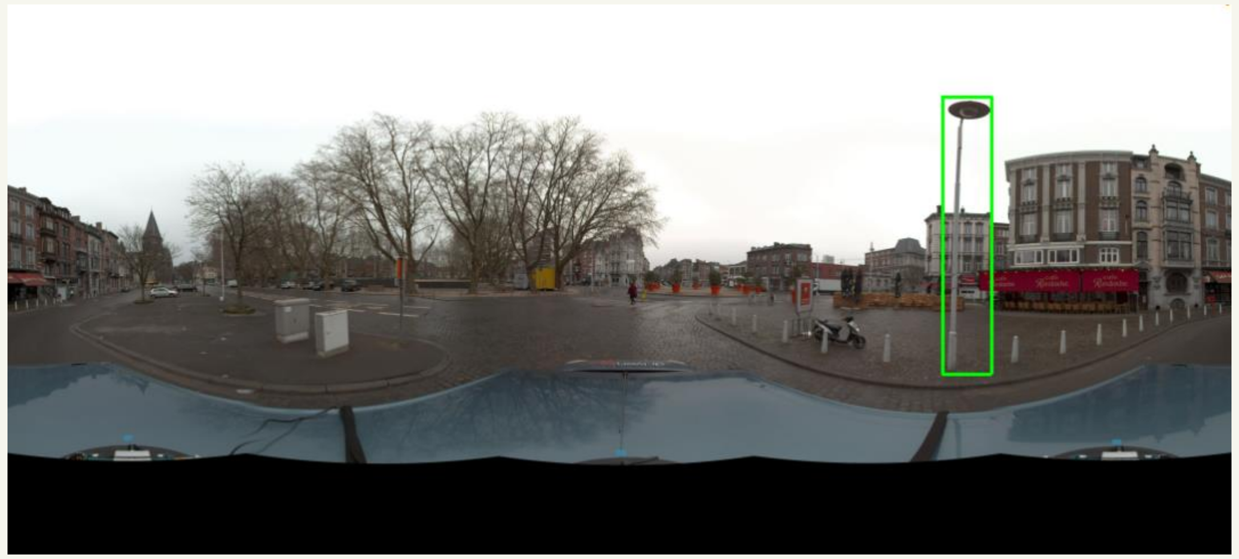
Direct Object Detection

	Bus Stop	Traffic light	Lamppost	Traffic signs
mIoU	0,850	0,780	0,810	0,830
Precision	0,974	0,885	0,853	0,880
Recall	0,909	0,896	0,838	0,875
mAP50	0,942	0,946	0,909	0,927
mAP50-95	0,685	0,737	0,648	0,552

Evaluation metrics for all classes

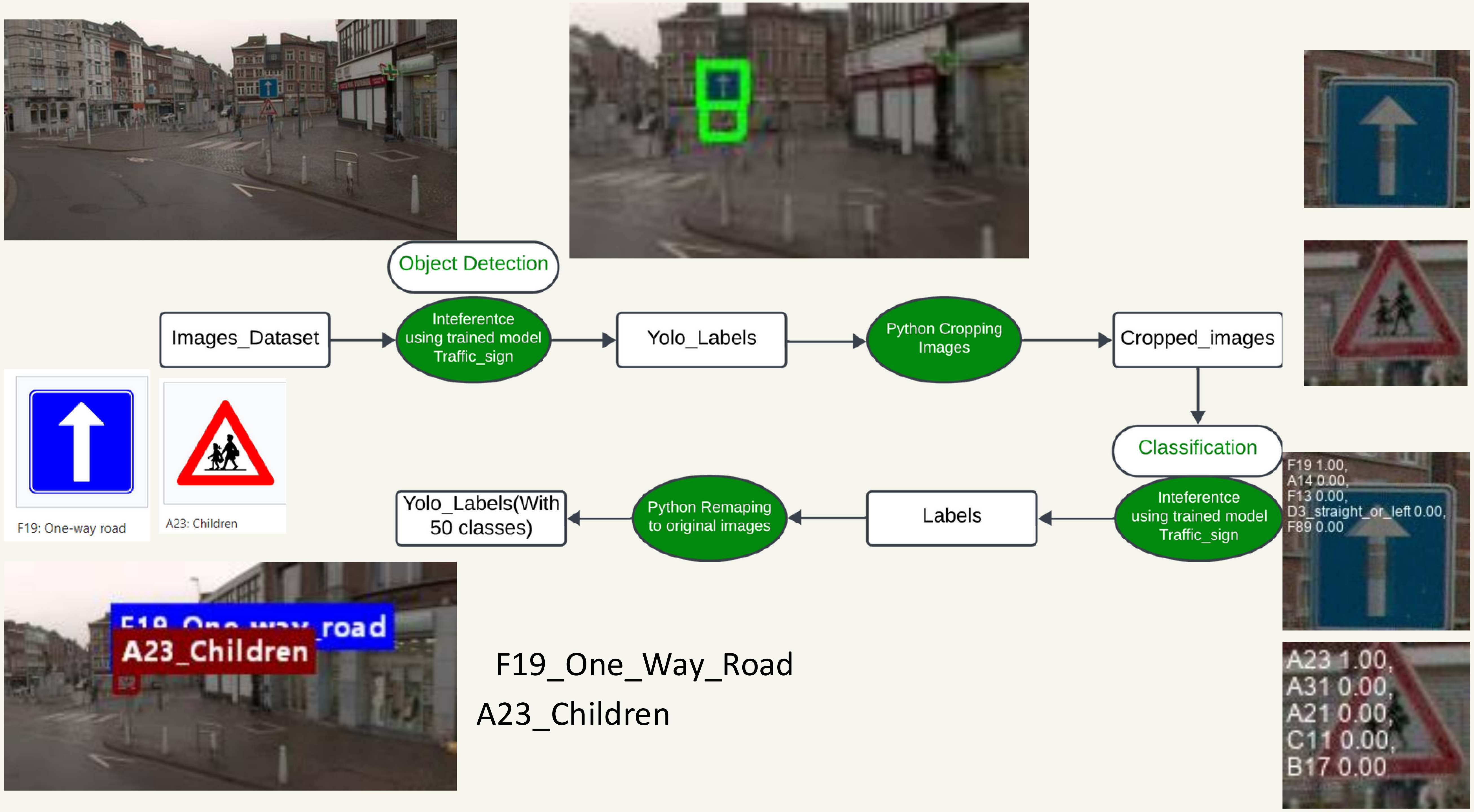
	Single	Building	Double
Presision	0,854	0,953	0,753
Recall	0,769	0,745	1,000
mAP50	0,863	0,903	0,962
mAP50-95	0,594	0,962	0,699

Evaluation metrics for lamppost subclasses



Object detection

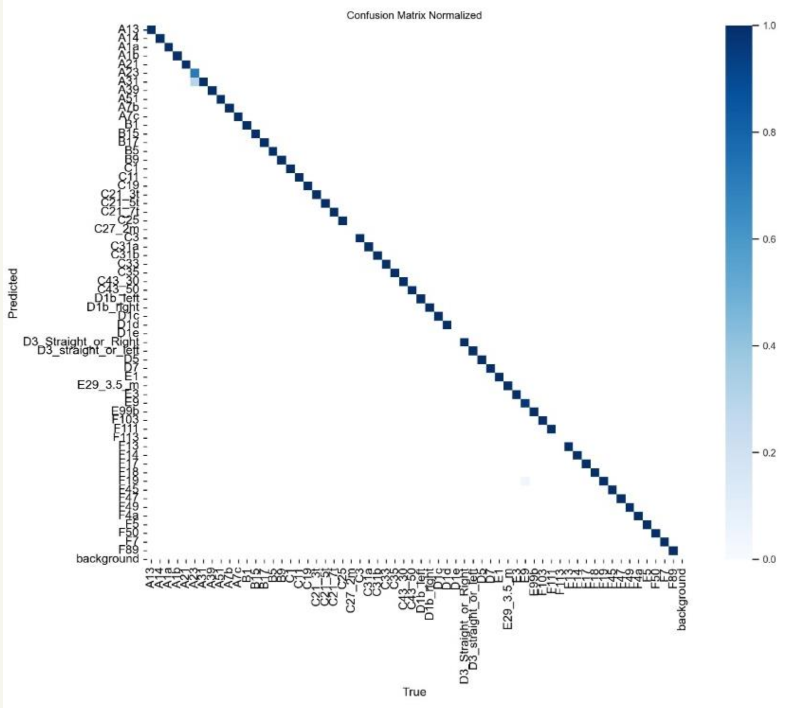
Cascaded Object Classification



Object detection

Cascaded Object Classification

Accuracy_Top_1	Accuracy_Top_5
0.99054	0.99369



C24b: No entry for drivers of vehicles transporting flammable or explosive goods, as defined by the minister of transportation

C24c: No entry for drivers of vehicles transporting polluting goods, as defined by the minister of transportation

C25: No entry for drivers of vehicles that exceed the combined length indicated on the sign

C27: No entry for drivers of vehicles that exceed the combined width indicated on the sign

C29: No entry for drivers of vehicles that exceed the combined height indicated on the sign

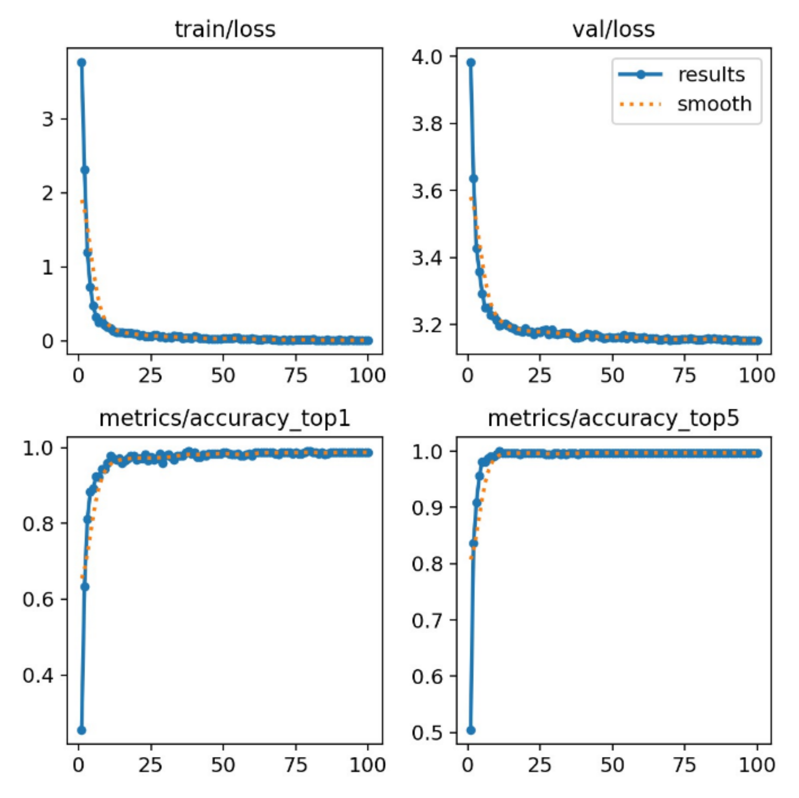
C31a: Left turn forbidden

C31b: Right turn forbidden

C33: U-turns forbidden

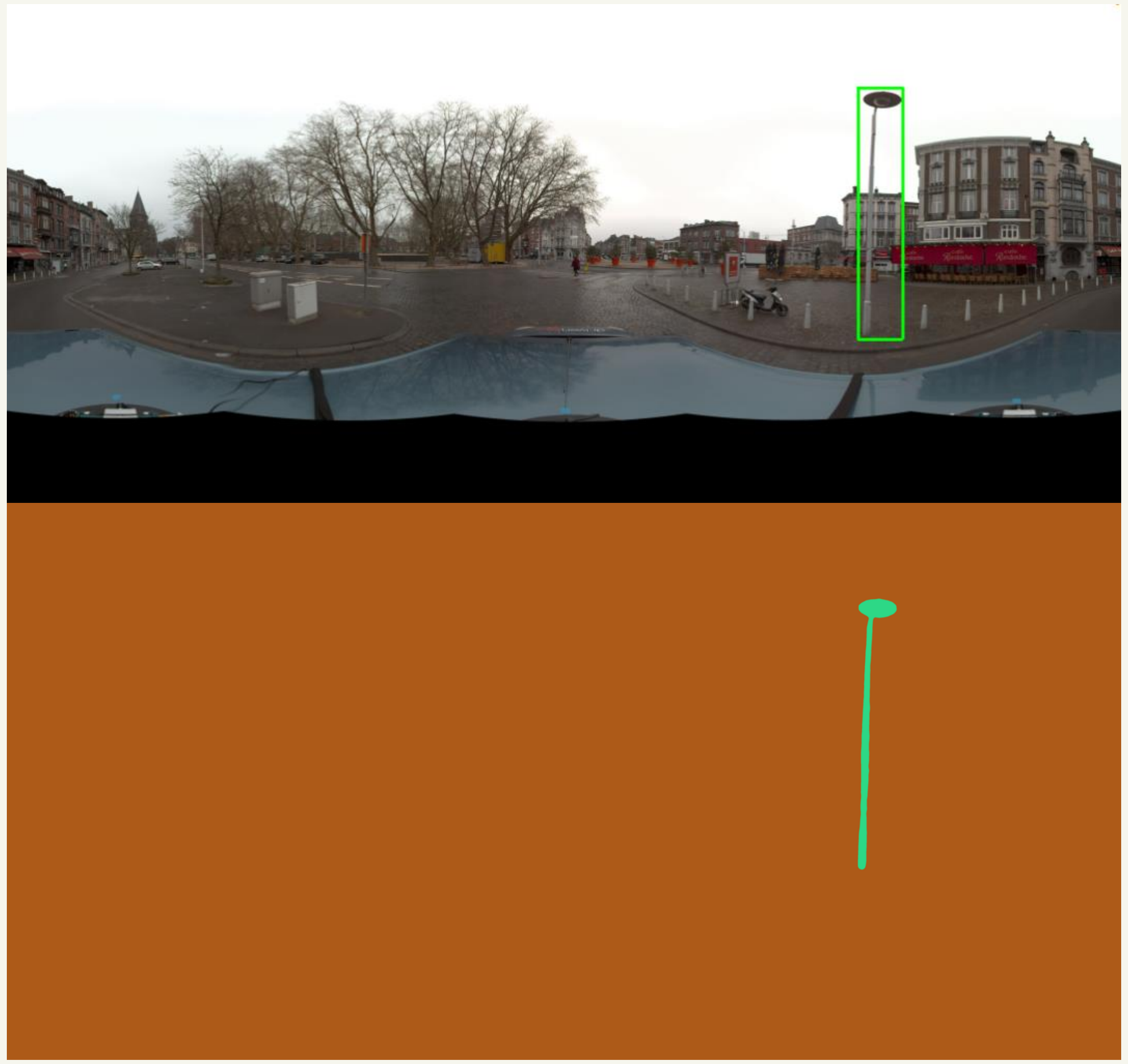
C35: Overtaking vehicles with more than two wheels and horse wagons forbidden

C37: End of the C35 restriction

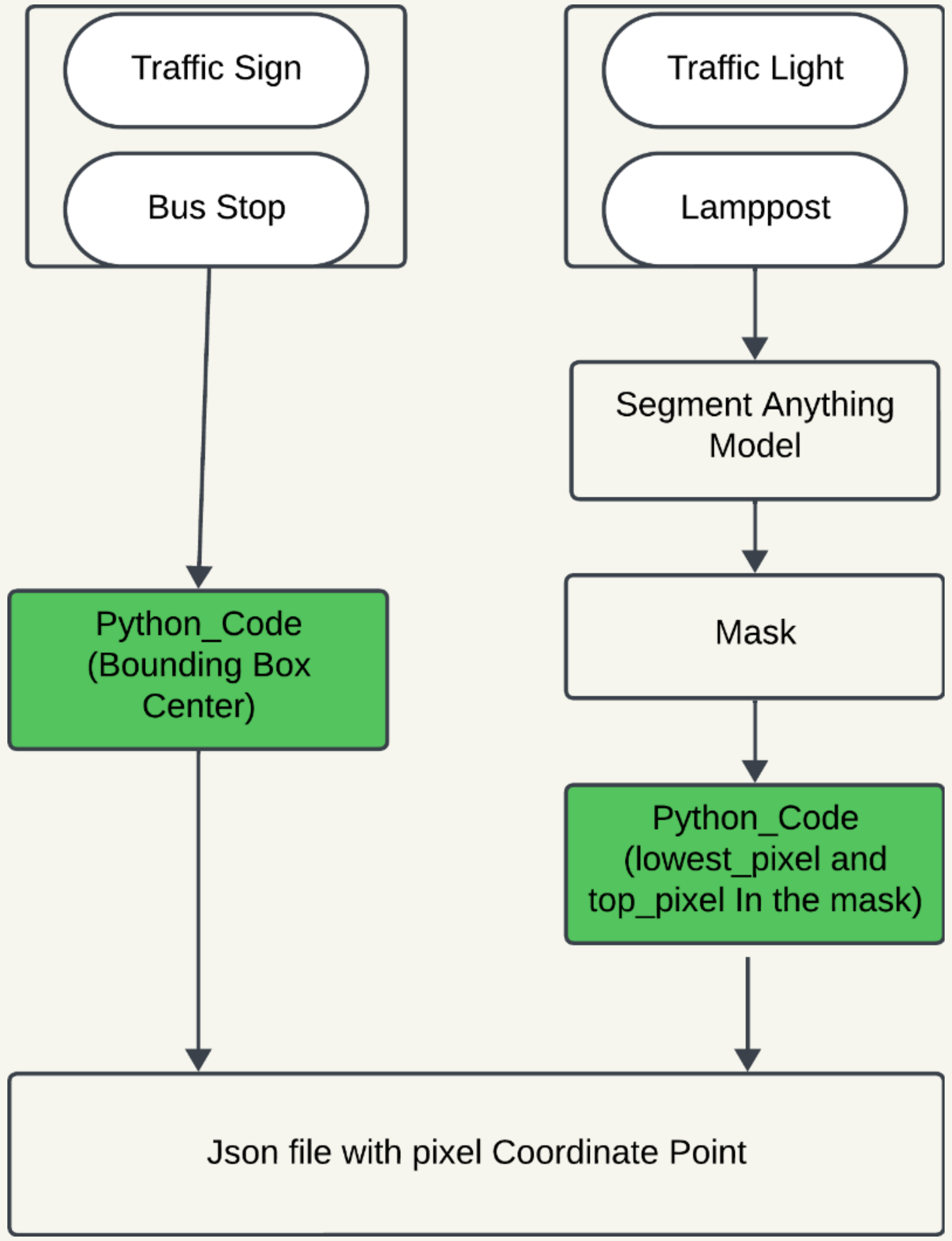


Positioning

Segment Anything Model (SAM) for image segmentation developed by Meta Research.



The use of the generated mask to identify the top and bottom pixels coordinate

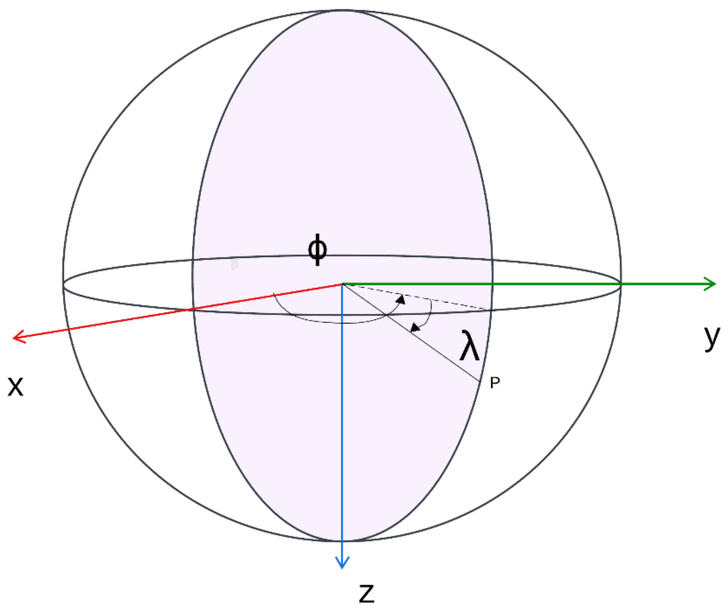


Positioning

Algorithm for calculating the line of bearing and vertical angle for a single object

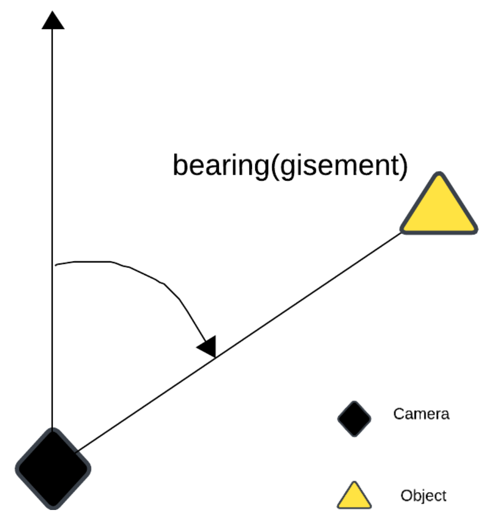
Pixel coordinates

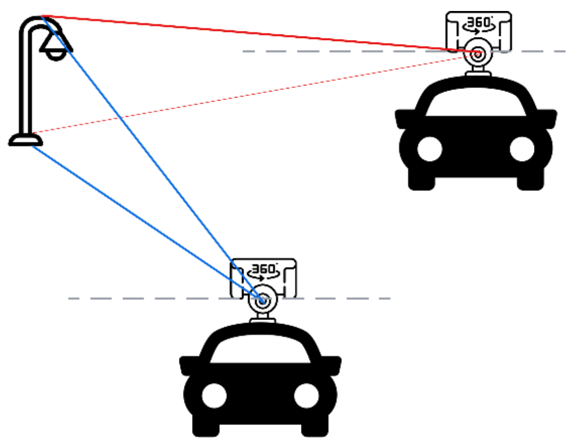
Polar coordinates

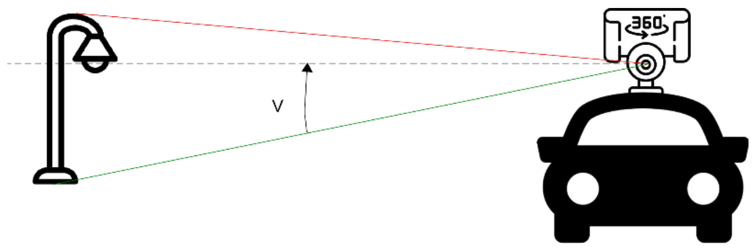


$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = sR \begin{bmatrix} x_c \\ y_c \\ -z_c \end{bmatrix} + \begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix}$$

The Y-axis represents the north in the projection.





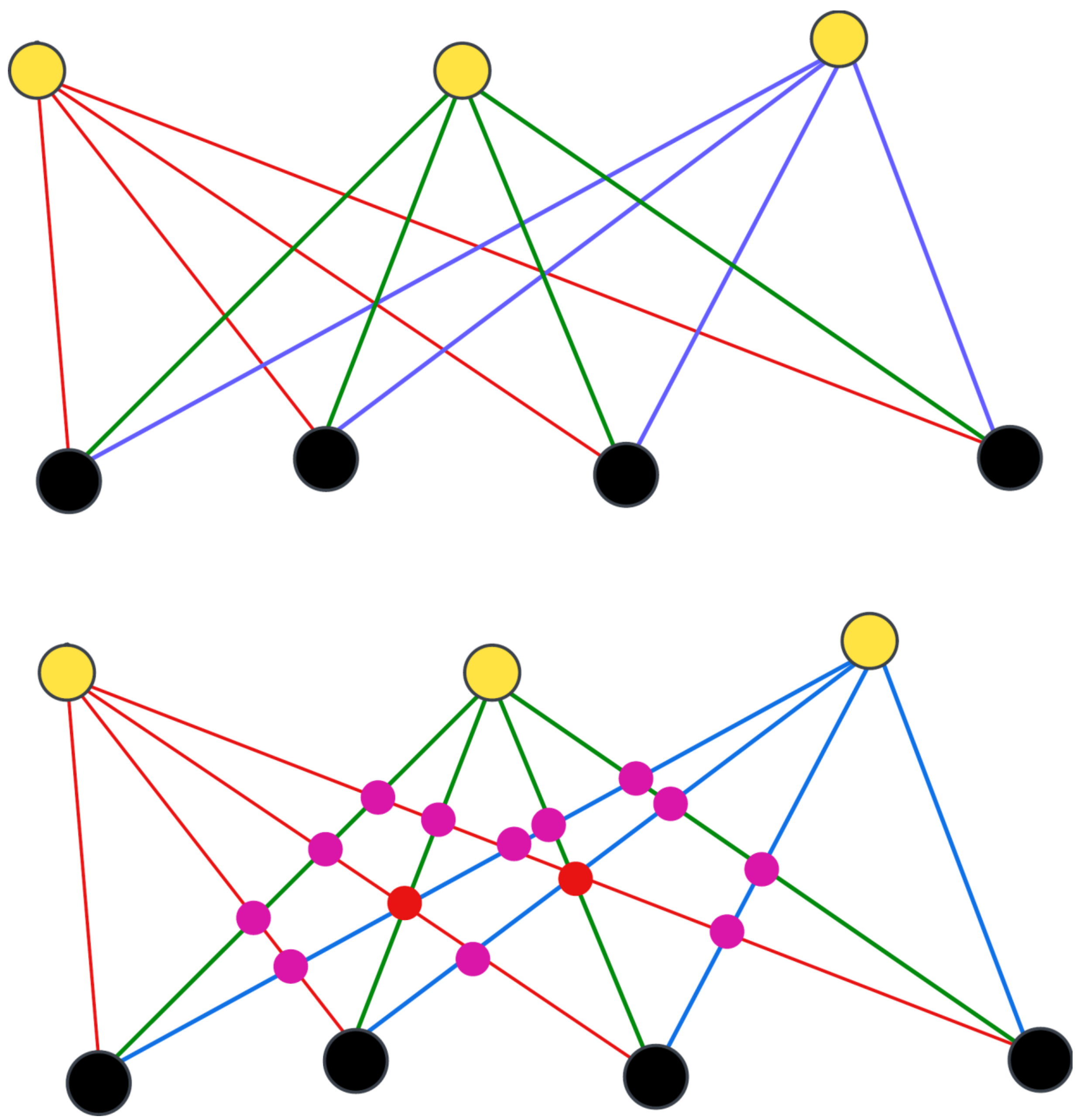


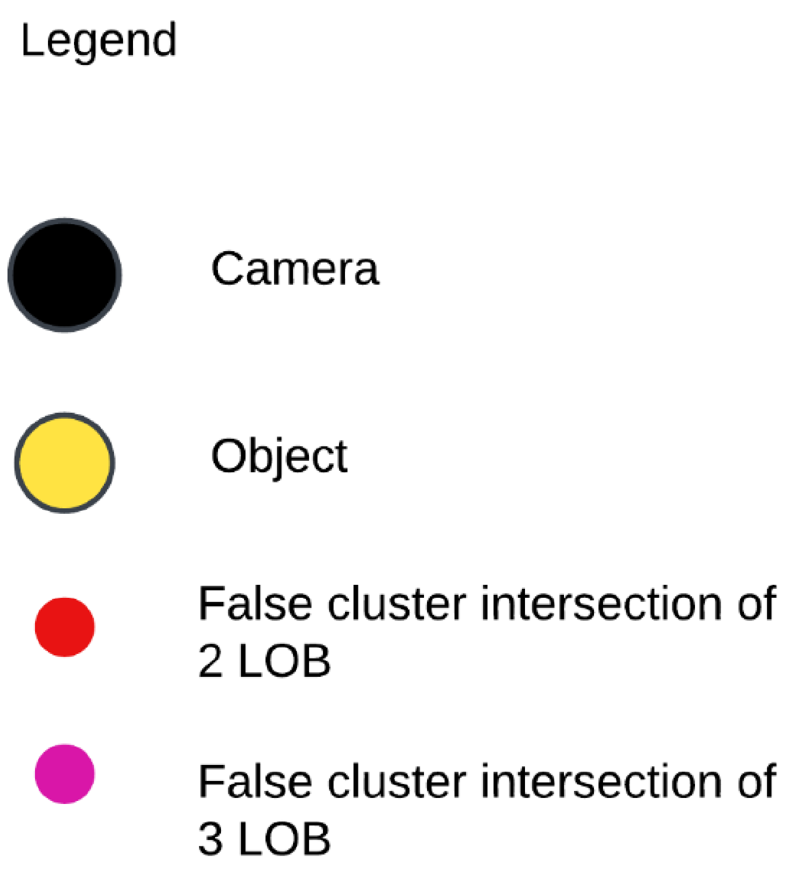
We calculate the intersection between lines of bearing of the same object.

Positioning

Extended Algorithm for Simultaneous Calculation of Lines of Bearing and Vertical Angles for Multiple Objects

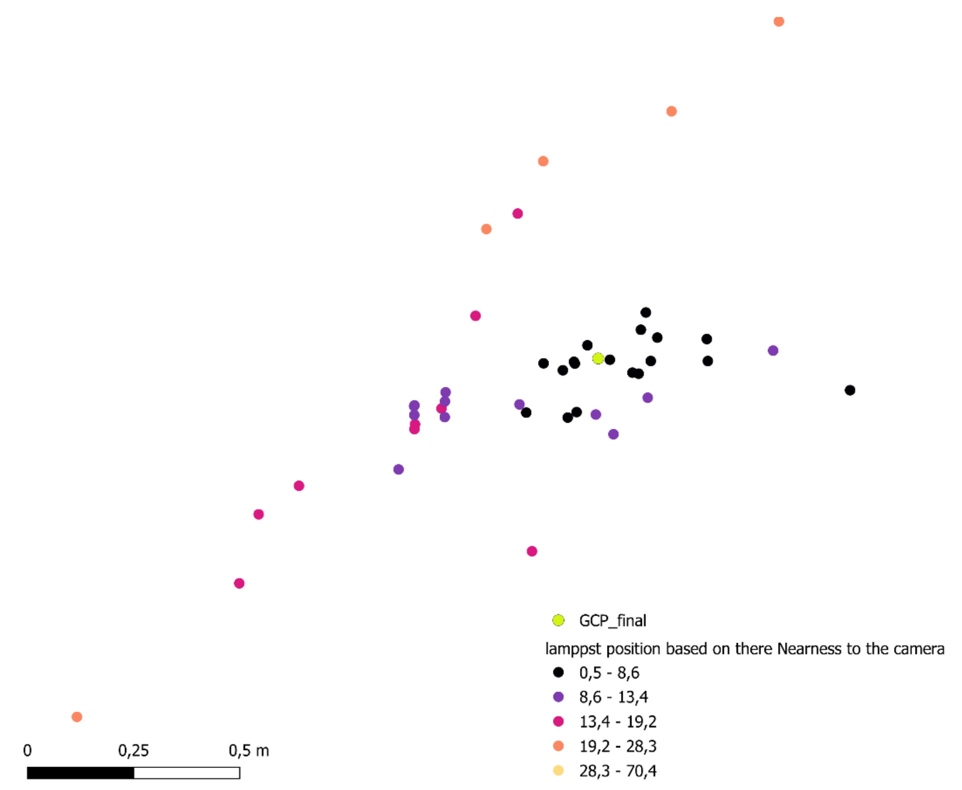
Li, G., Lu, X., Lin, B., Zhou, L., Lv, G., 2022. Automatic Positioning of Street Objects Based on Self-Adaptive Constrained Line of Bearing from Street-View Images. ISPRS International Journal of Geo-Information, 11.





Positioning

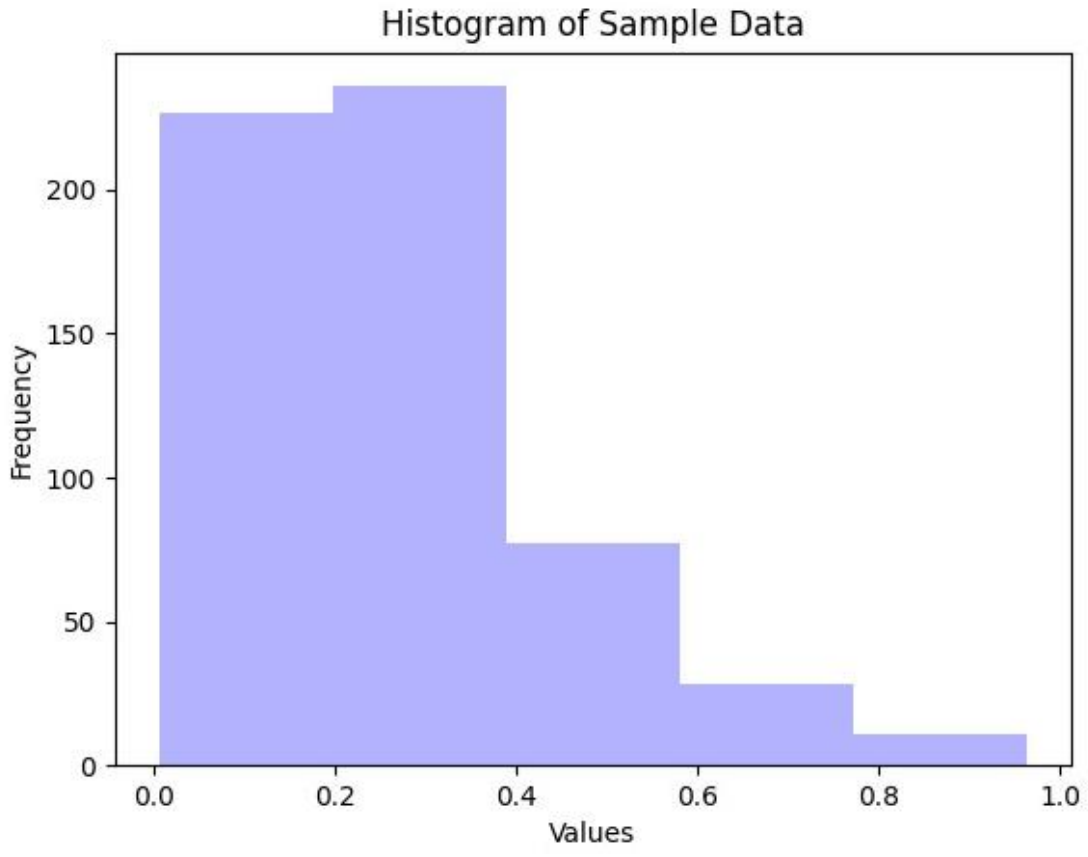
Centroid Calculation with Proximity Filtering



We observed that points captured from a distant camera position significantly deviate from the ground control points. As a result, we decided to eliminate the distant points and, based on a threshold, retain only the closer points.

Precision of our detection compared to PICC DATA

Metric	Value
Total Numbers of point	579
Mean error	0.27 m
RMSE	0.32 m



Positioning

Liège

Object Type	Number of instance
Single Lamppost	933
Bat Lamppost	119
Bus Stop	34
Traffic Light	146
All Traffic sign combined	736

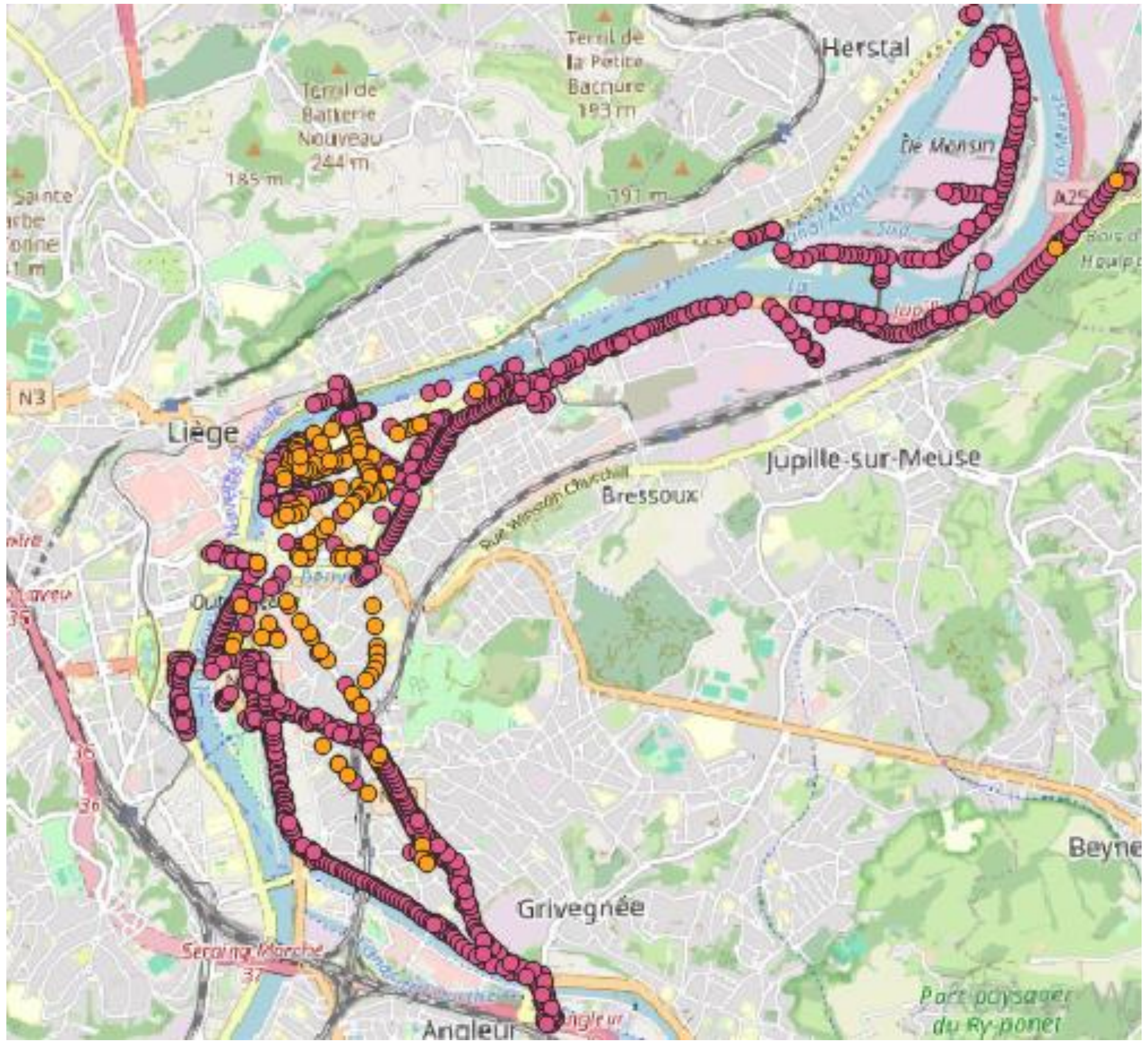
Arlon

Object Type	Number of instances
Single Lamppost	34
Bat Lamppost	18
Double Lamppost	3
Traffic Light	13
All Traffic sign combined	31

- Context
- Motivations
- Objectives
- **Imagery-based Approach**
- Camera-LiDAR Approach
- Modeling Approach
- Discussion
- Conclusion and Perspectives

Positioning

Results: Lamppost



- cluster point of single lamppost
- cluster point of wall-mounted street light

- Context
- Motivations
- Objectives
- **Imagery-based Approach**
- Camera-LiDAR Approach
- Modeling Approach
- Discussion
- Conclusion and Perspectives

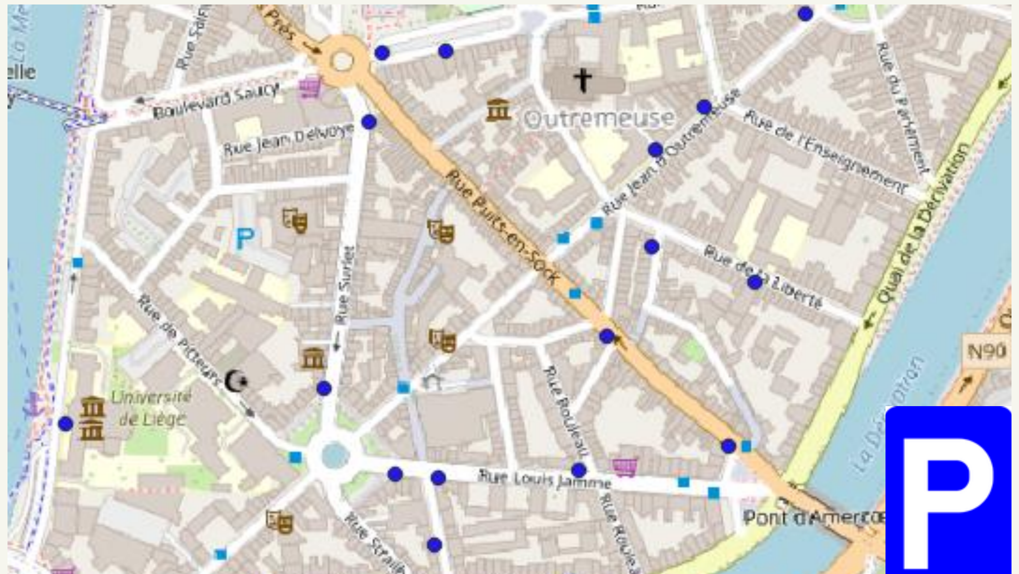
Positioning

Results:

Lamppost

Traffic Light

Traffic sign
60 classes
detected

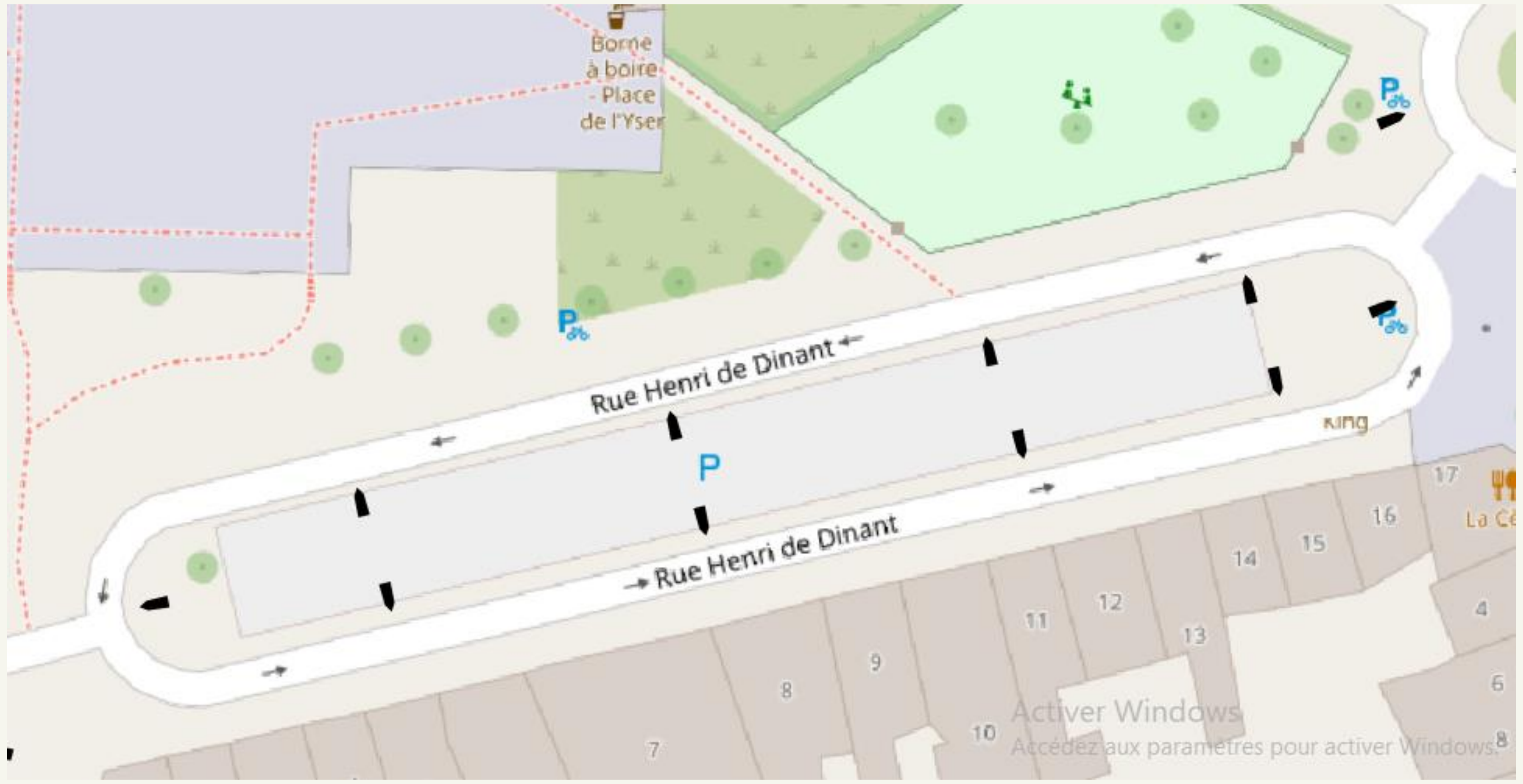


Context
Motivations
Objectives
Imagery-based Approach
Camera-LiDAR Approach
Modeling Approach
Discussion
Conclusion and Perspectives

Feature extraction

Object Orientation and Height

We calculate the object **orientation** as the angle (**azimuth**) between a paramtric model where the object is parallel to the y-axis and the track/road axes. The object position relatively to the track line (left or righ) is also considered. We also calculate each object **height** as the difference between elevations of the top and the bottom points.

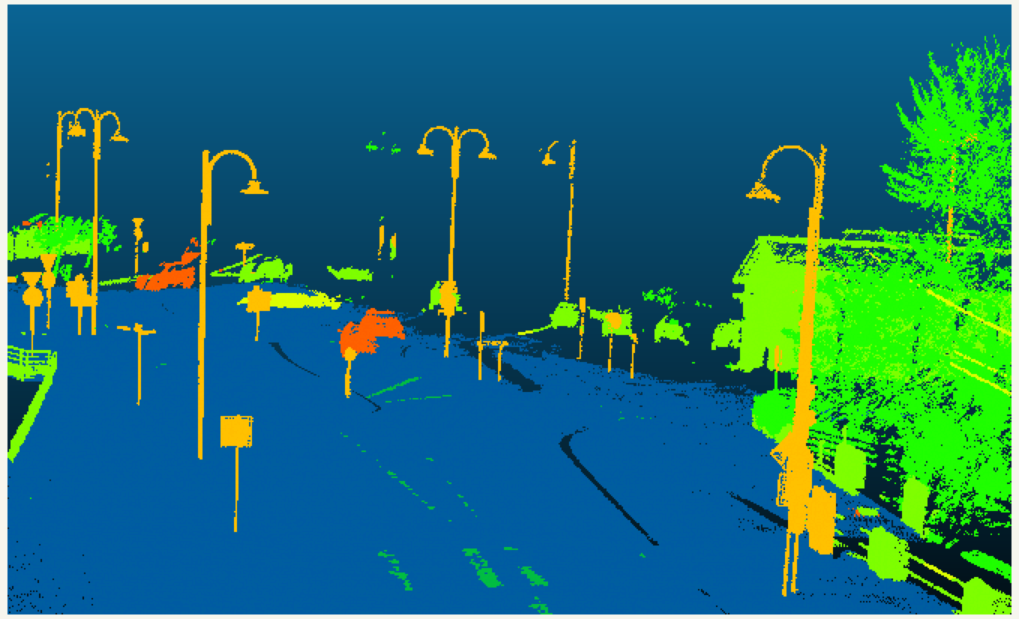


Camera-LiDAR Approach

Data Collection

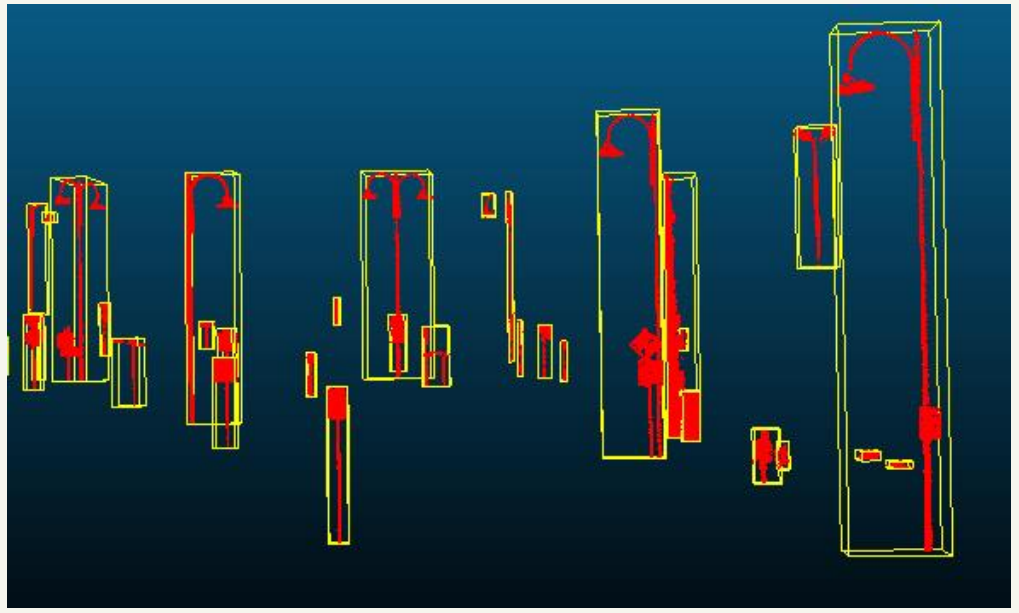
Sementic Segmentation

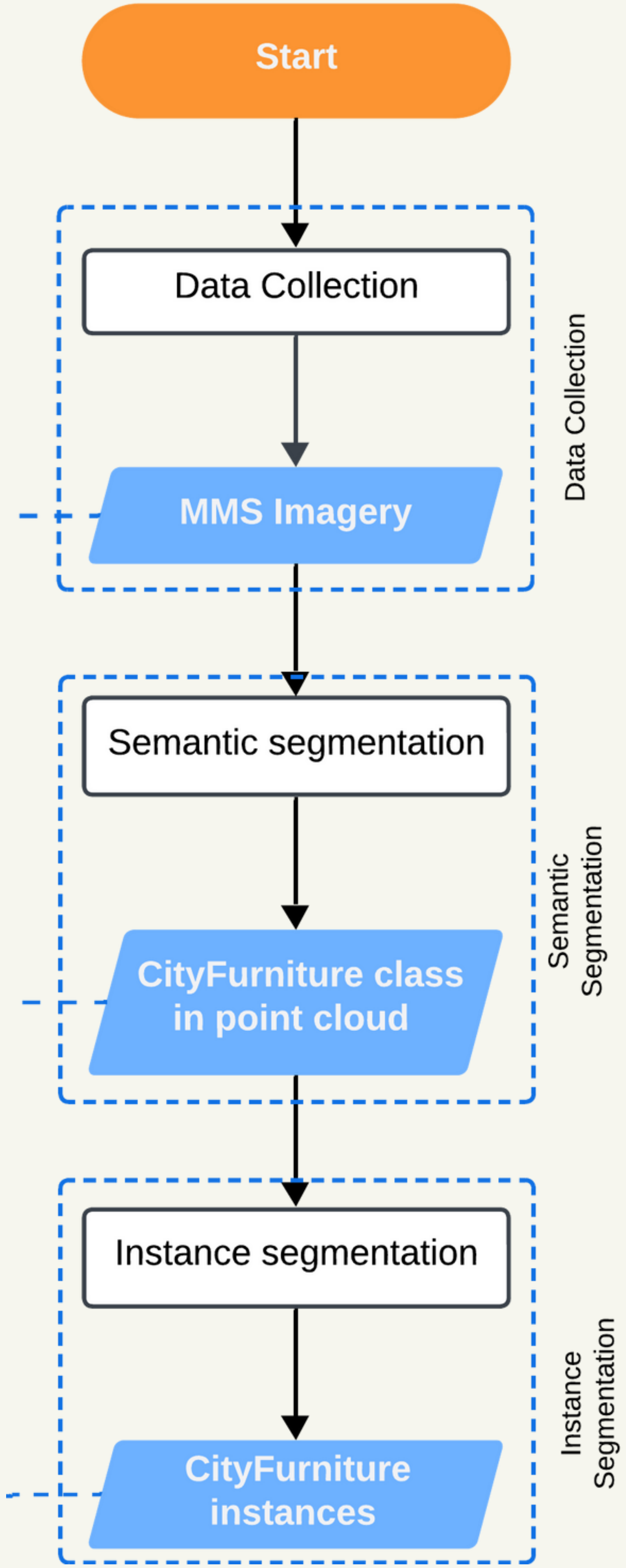
KPConv



Instance Segmentation

Label Connected Component (LCC)

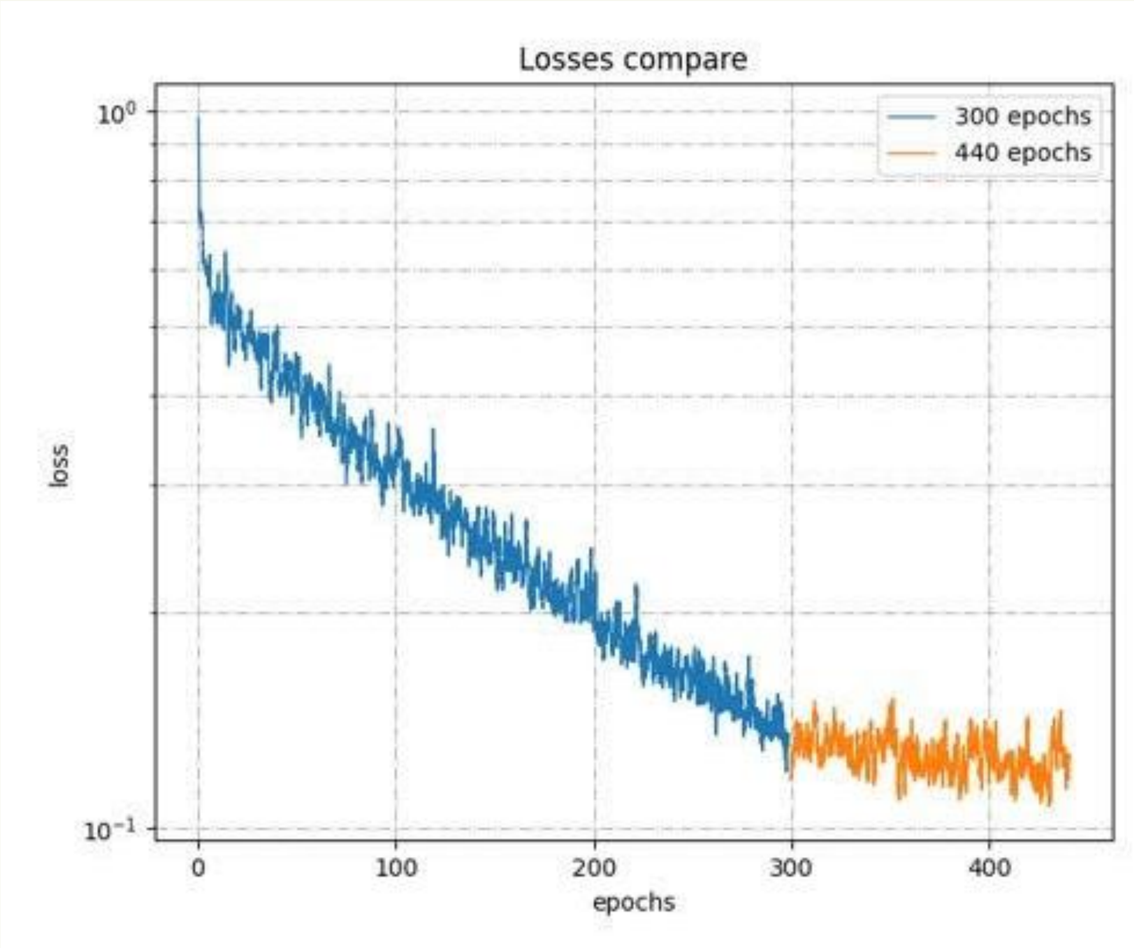


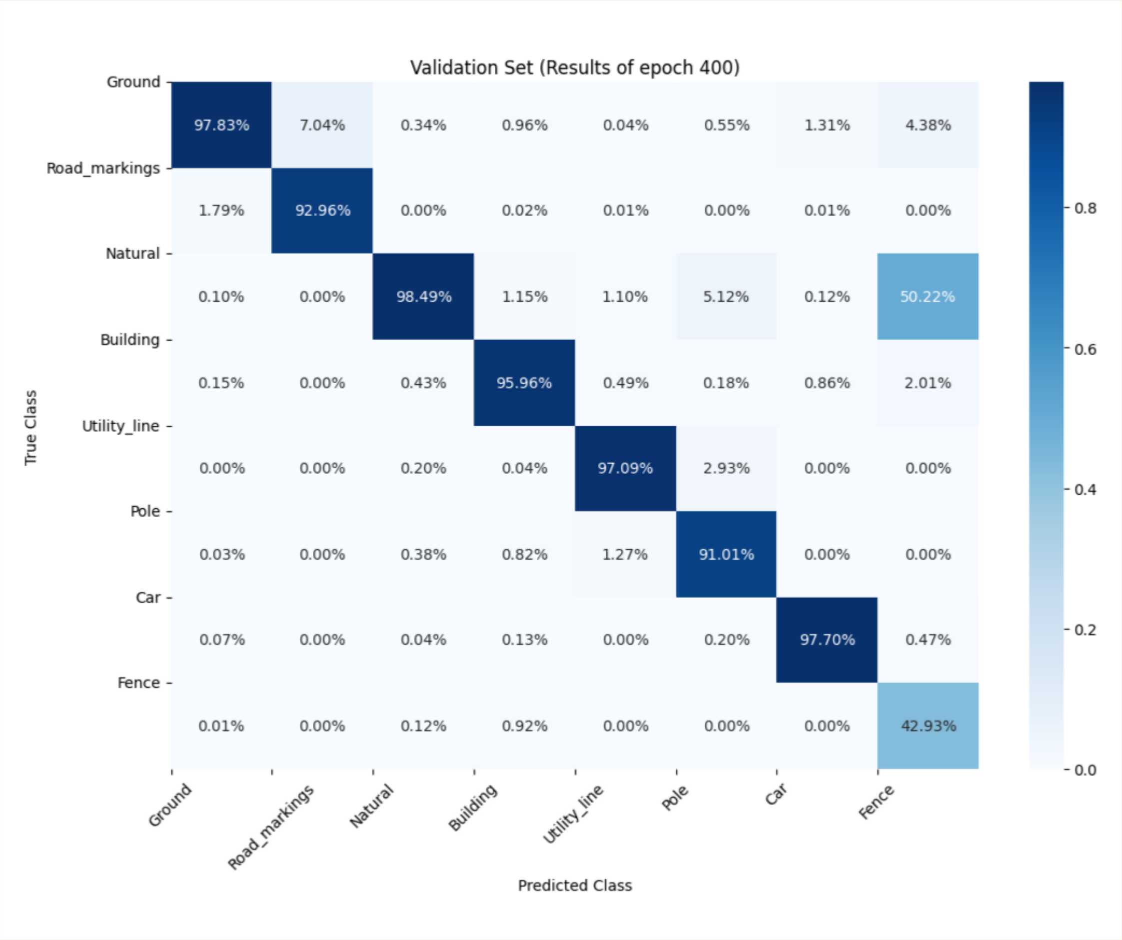


Semantic segmentation

KPConv, or Kernel Point Convolution, is an architecture for processing 3D point clouds directly, without converting them into a grid or other structure. It uses points in space (kernel points) to apply convolution operations directly on the point cloud.

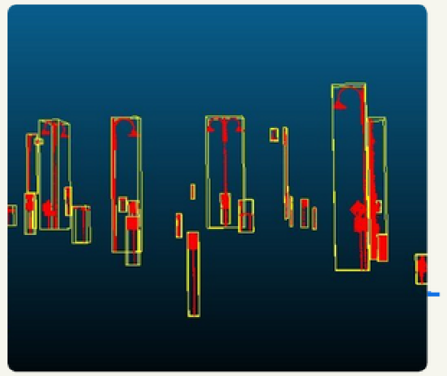
Our KPconv model was trained on *Toronto 3D* dataset with the following performance:

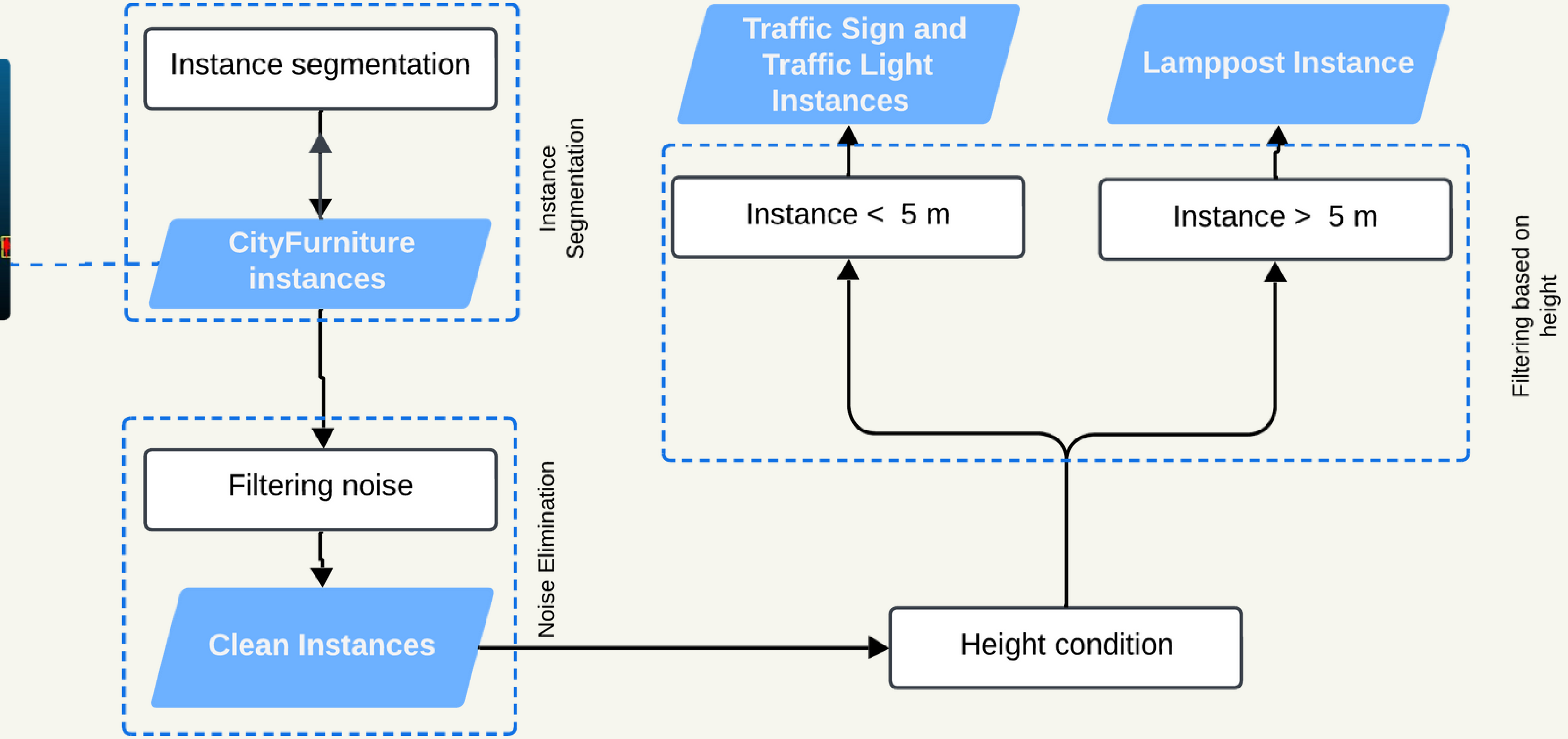




OA	mIoU	Ground	Road marking	Natural	Building	Cable	Pole	Car	Fence
94.7%	79.0%	96.6%	61.7%	95.1%	80.1%	82.5%	78.4%	87.1%	41.8%

Classification





Classification

Image-Based Classification

Traffic signs , Traffic light

↓

Reprojection 3D bounding Box into the 4 Closest images

↓

Cropped images

↓

Grounding Dino to crop the traffic sign

↓

Yolov8 classification model (traffic light , traffic sign)

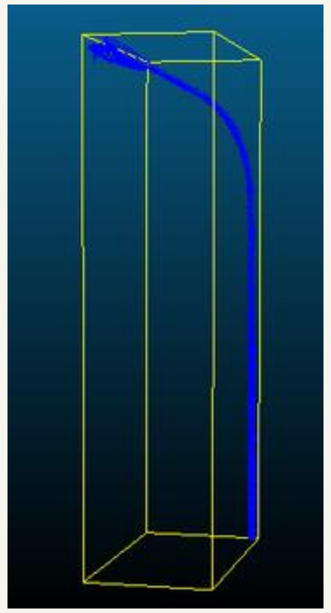
↓

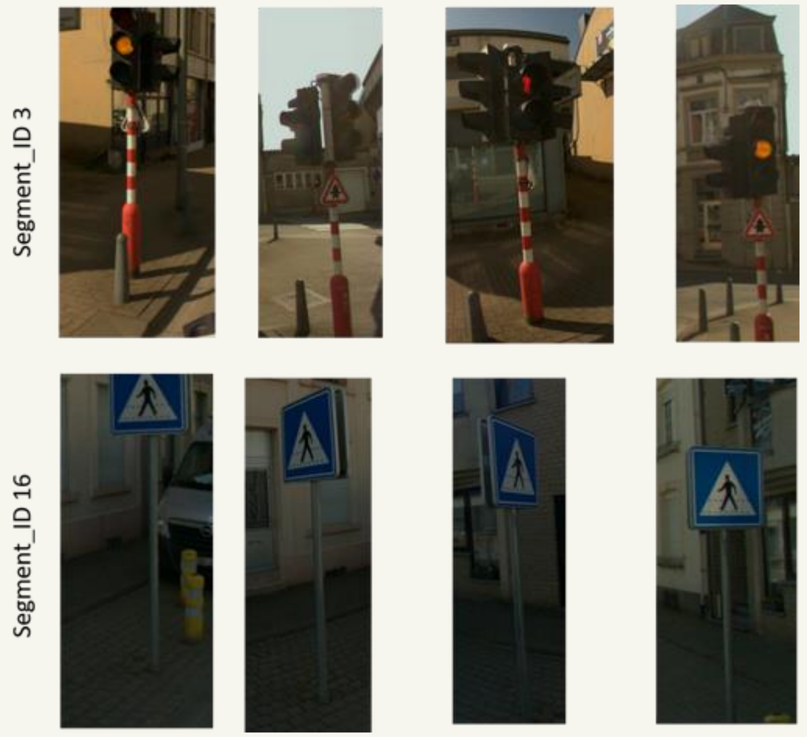
Geoparquet file :

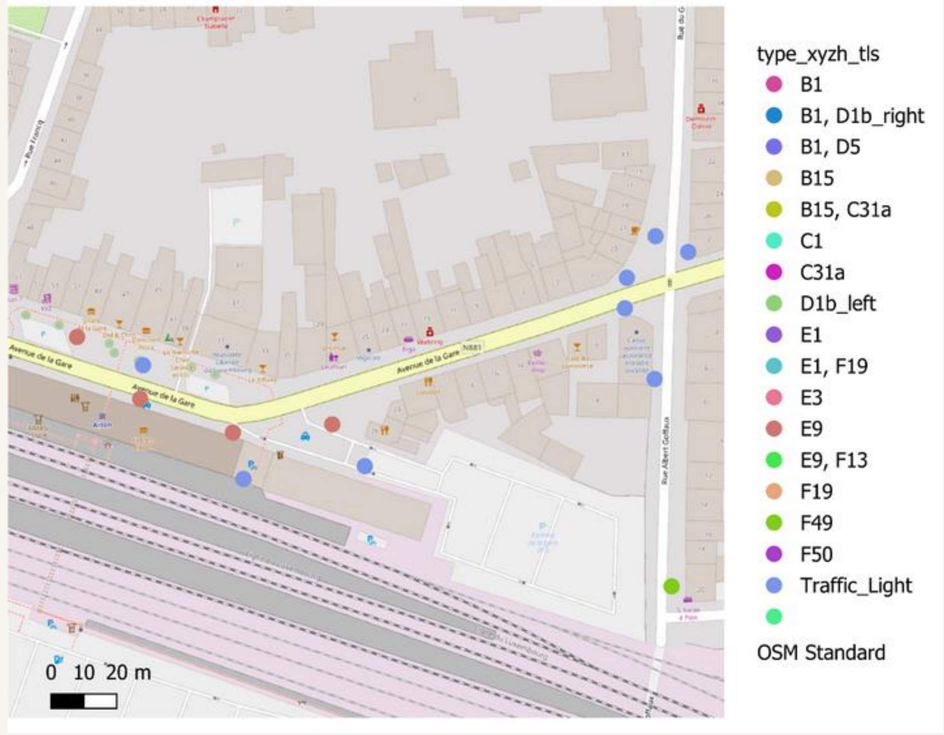
↓

id	x	y	z	h	cityfurniture_type

1_Reprojection 3D bounding box into 2D image

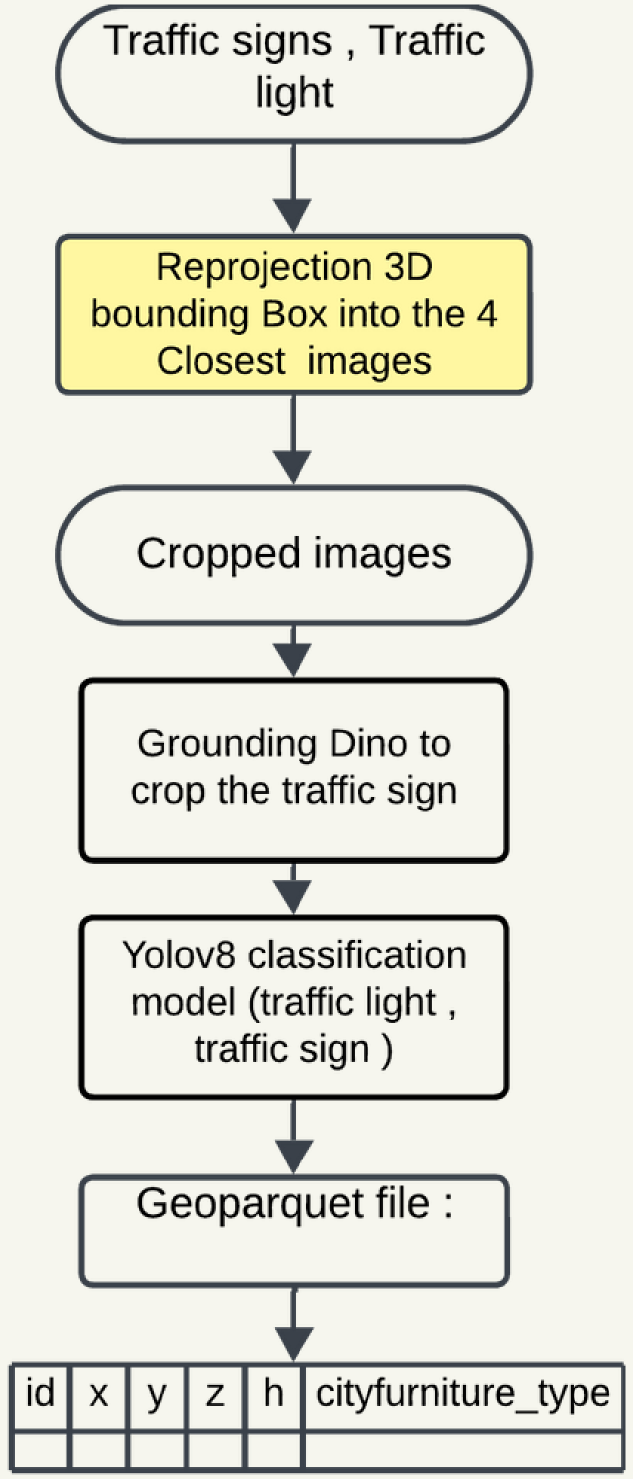




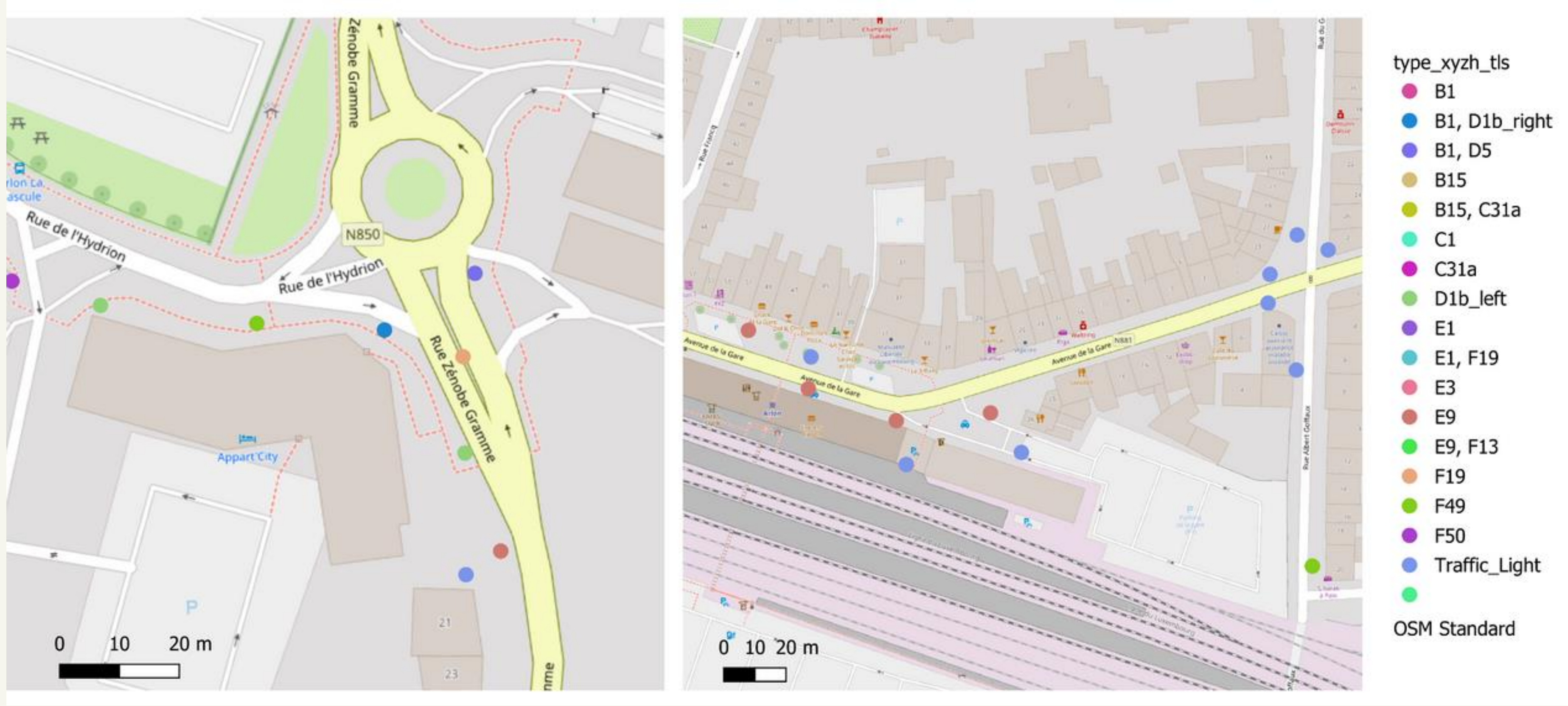


Classification

Image-Based Classification

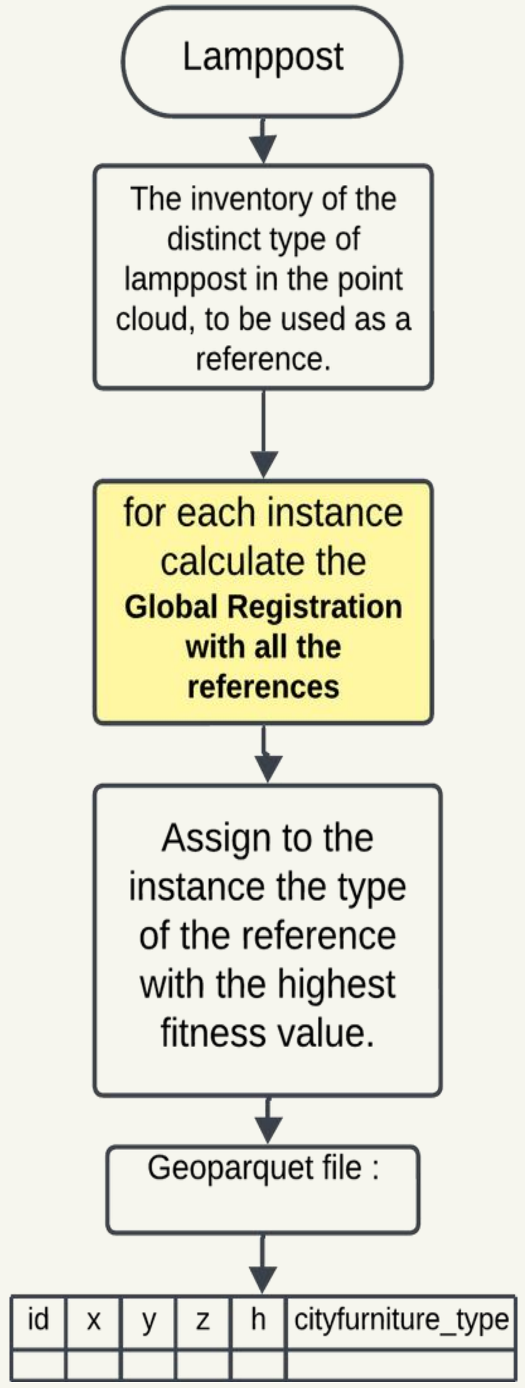


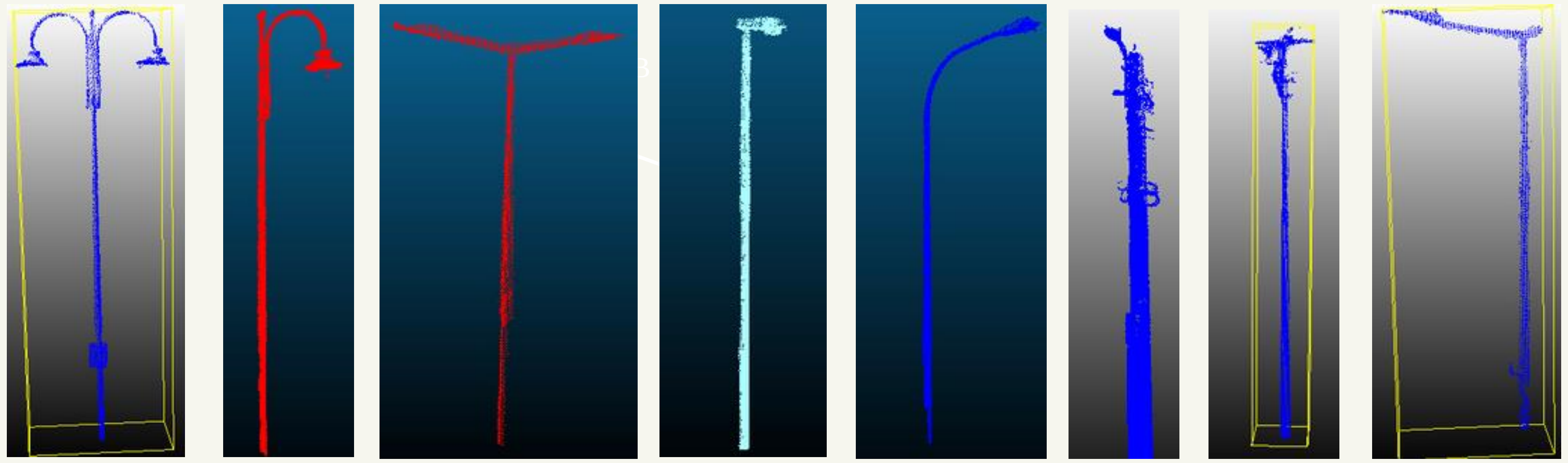
	segment_id	final_cl	X	Y	Z	height	__index_level_0__
1	2	Traffic_Light	254139,0535000...	42015,64650122...	392,803	3,016000000000...	0
2	12	B1, D1b_right	253681,0595000...	41946,58050122...	390,003	3,324000000000...	16
3	14	F49	253834,7705000...	41792,48350122...	380,074	2,766999999999...	17
4	25	E9	254049,9455000...	41980,3955012207	390,879	2,935999999999...	18
5	27	E3	253652,9315000...	41804,8765012207	380,514	3,278999999999...	19
6	28	E9, F13	254187,1685000...	42453,09150122...	416,802	2,969999999999...	20
7	29	Traffic_Light	254136,9475000...	42395,56350122...	415,6	3,06899999999996	1
8	30	B15	254196,2815000...	42465,34150122...	416,647	3,305000000000...	21
9	33	Traffic_Light	254139,6545000...	42024,8765012207	393,036	3,086000000000...	2



Classification

LiDAR-Based Classification



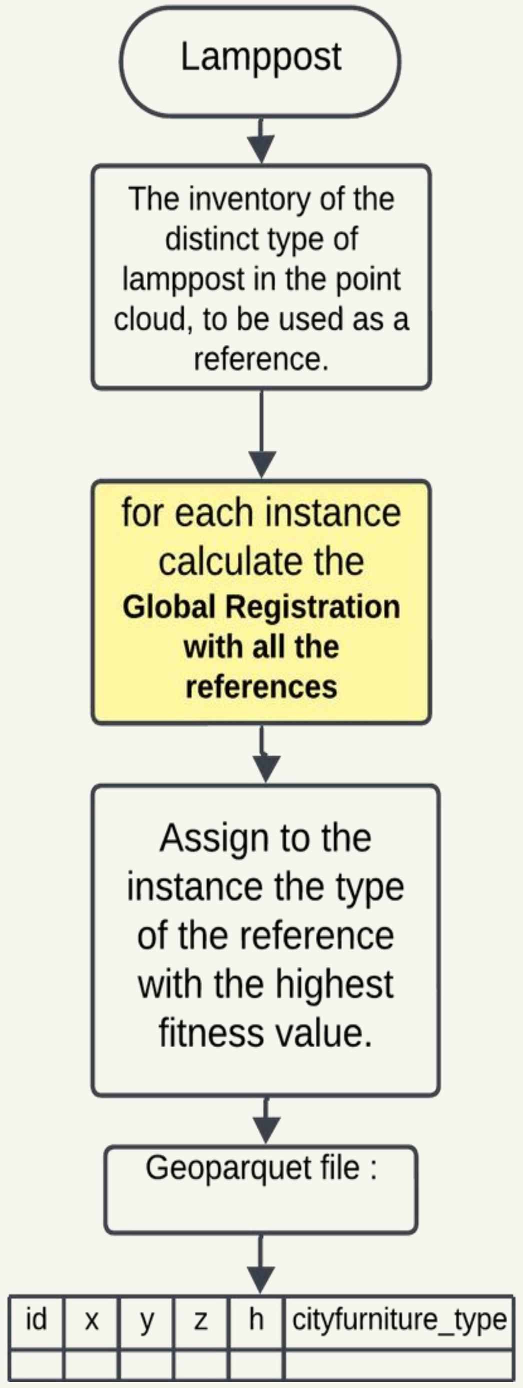


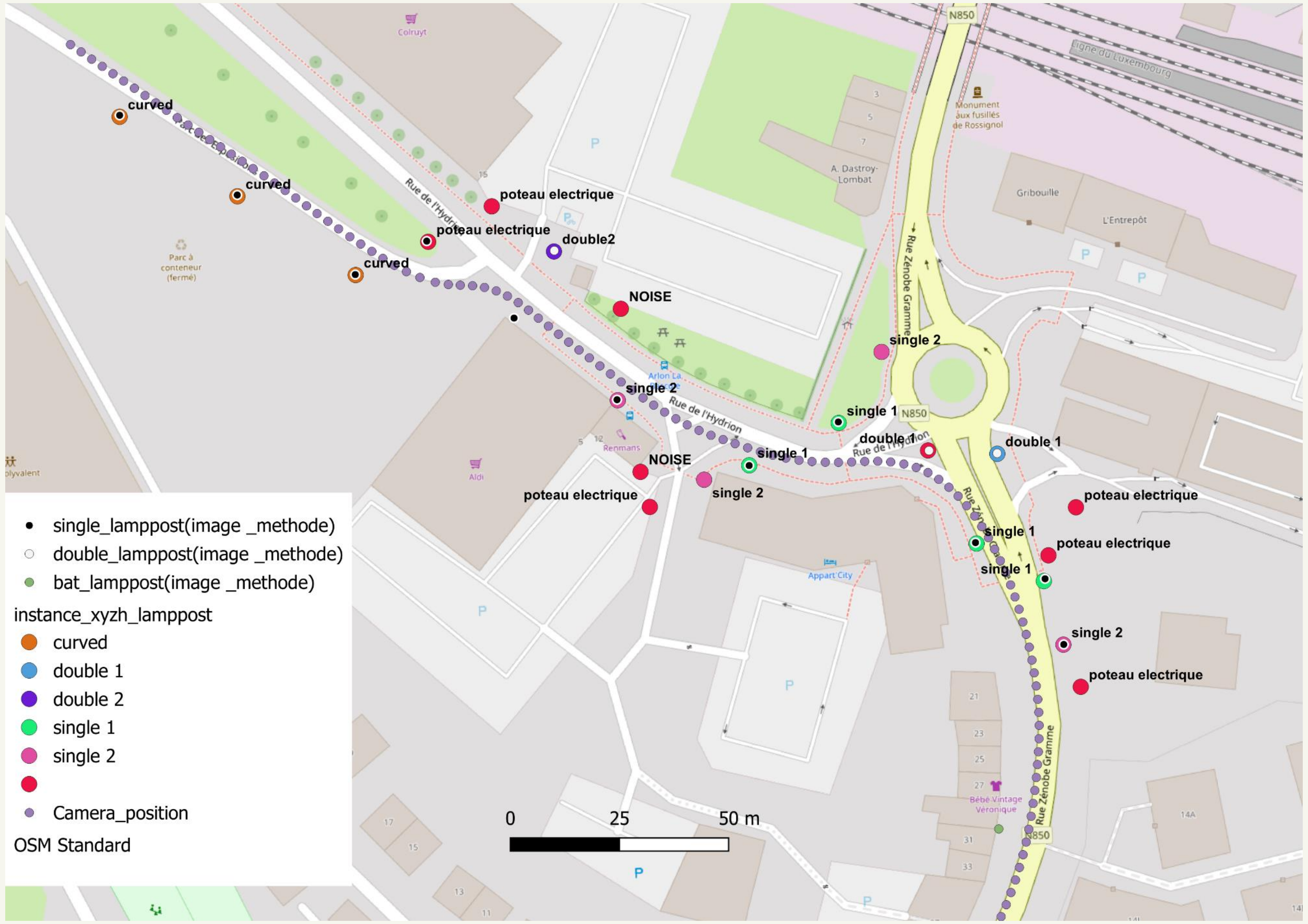
0 1 2 3 4 5 6
7

- Among the detected lampposts, certain classes, such as electric poles, are particularly challenging to classify.
- Noise in the data suggests that the point cloud requires further filtering and stricter constraints.
- The global registration method successfully identified the most distinct classes (0, 1, 2, 3, 4) with a precision rate of 0.91, assuming these were the only types present initially.
- However, the success of this method heavily depends on the elimination of noise and the presence of clearly defined classes from the outset. Before manual filtering was applied to our dataset, the success rate dropped to 50%.

Classification

LiDAR-Based Classification





LIÈGE université
GeoScITY

3D Modeling

CityJSON

CityJSON Format

- For each CityObject we should at least define the type and the geometry:
  - The type the geometry Object in our case is CityFurniture.
  - The geometry could be an array of the geometry object defined by CityJSON:

```
{
  "type": "CityFurniture",
  "geometry": [
    {
      "type": "GeometryInstance",
      "template": 0,
      "boundaries": [372],
      "transformationMatrix": [
        2.0, 0.0, 0.0, 0.0,
        0.0, 2.0, 0.0, 0.0,
        0.0, 0.0, 2.0, 0.0,
        0.0, 0.0, 0.0, 1.0
      ]
    }
  ]
}
```

Given the **repetitive nature** of the city furniture in our model ,we have chosen to utilize the **geometry template** in CityJSON.

Geometry Instance Definition

```
"geometry-templates": {
  "templates": [
    {
      "type": "MultiSurface",
      "lod": "2.1",
      "boundaries": [
        [[0, 3, 2, 1]],
        .......
      ]
    },
    {
      .......
    }
  ],
  "vertices-templates": [
    [0.0, 0.5, 0.0],
    ...
  ]
}
```

- To use the geometry template we should first define the geometry template contenant les different template:Traffic light , traffic sign , lamppost and bus stop.
- Each “Cityobject” has the right to use only one geometry object as their geometry .

Texture and material Definition

For each template, we also define the texture and material using an Appearance Object. The Appearance Object serves as a reference, allowing us to consistently apply the same texture or material whenever needed in a geometry object.

```
"appearance": {
  "default-theme-material": "default",
  "default-theme-texture": "default",
  "materials": [ ···
  ],
  "textures": [
    {
      "image": "E:/PFE_Roadmap/_5_Modeling/Traffic_sign/With_texture/traffic_sign_be/F17.png",
      "type": "PNG"
    }
  ],
  "vertices-texture": [ ···
  ]
}
```

The definition of the Appearance as an object

```
"templates": [
  {
    "boundaries": [ ···
    ],
    "lod": "2",
    "material": { ···
    },
    "texture": { ···
    },
    "type": "MultiSurface"
  }
```

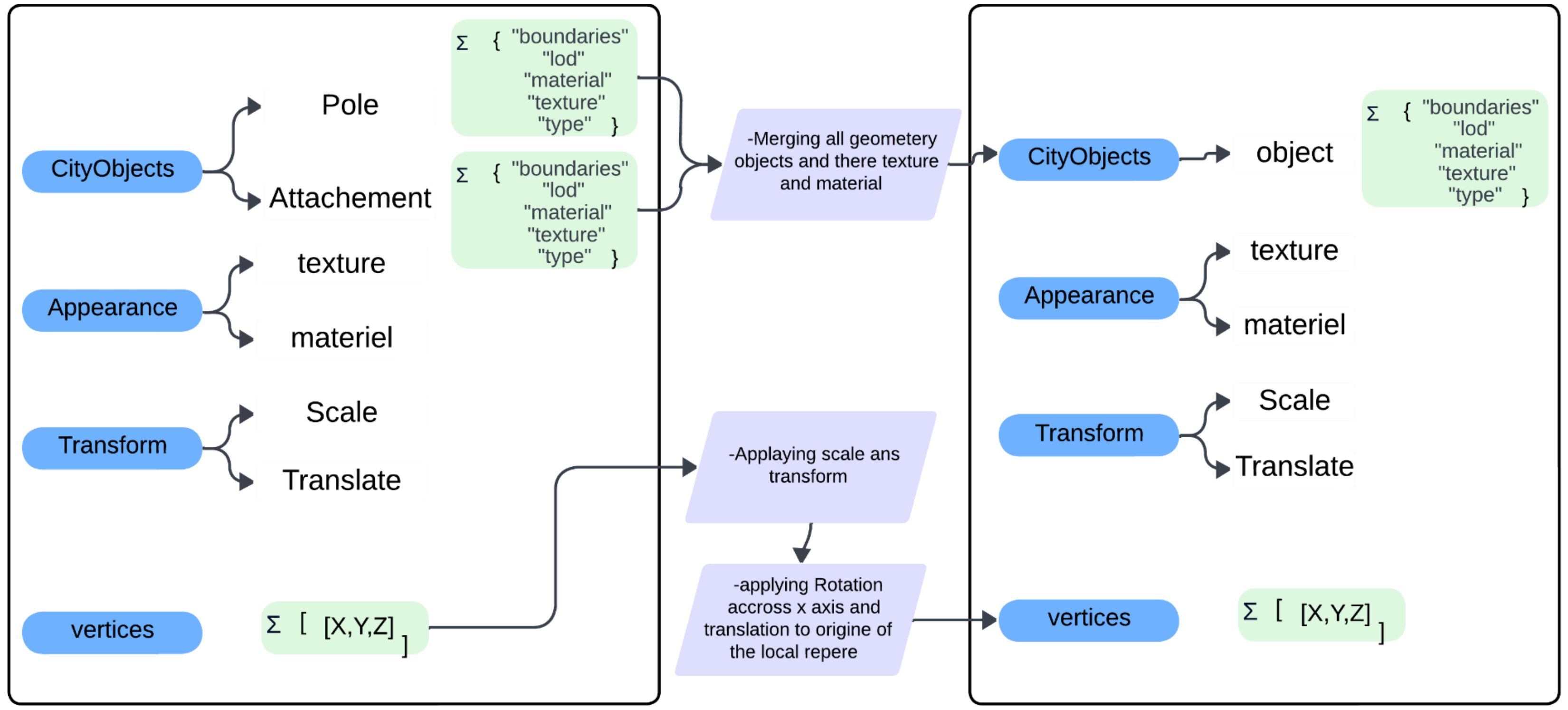
The reference to textures and materials is made through the Appearance definition.

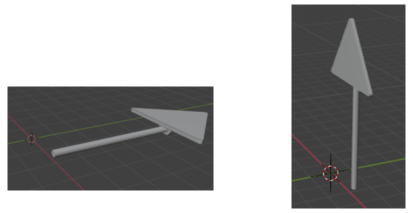
Model handling

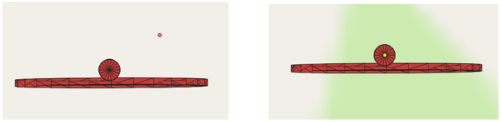
1. **Object Transformation to CityJSON Format**:

Conversion: The first step involves transforming your 3D objects from their native formats (such as .max, .fbx, or .obj) into the CityJSON format, specifically converting them into vertices and boundaries.

2. **Merge geometries to one geometry object (Adapt the model to be used as a geometry template)**

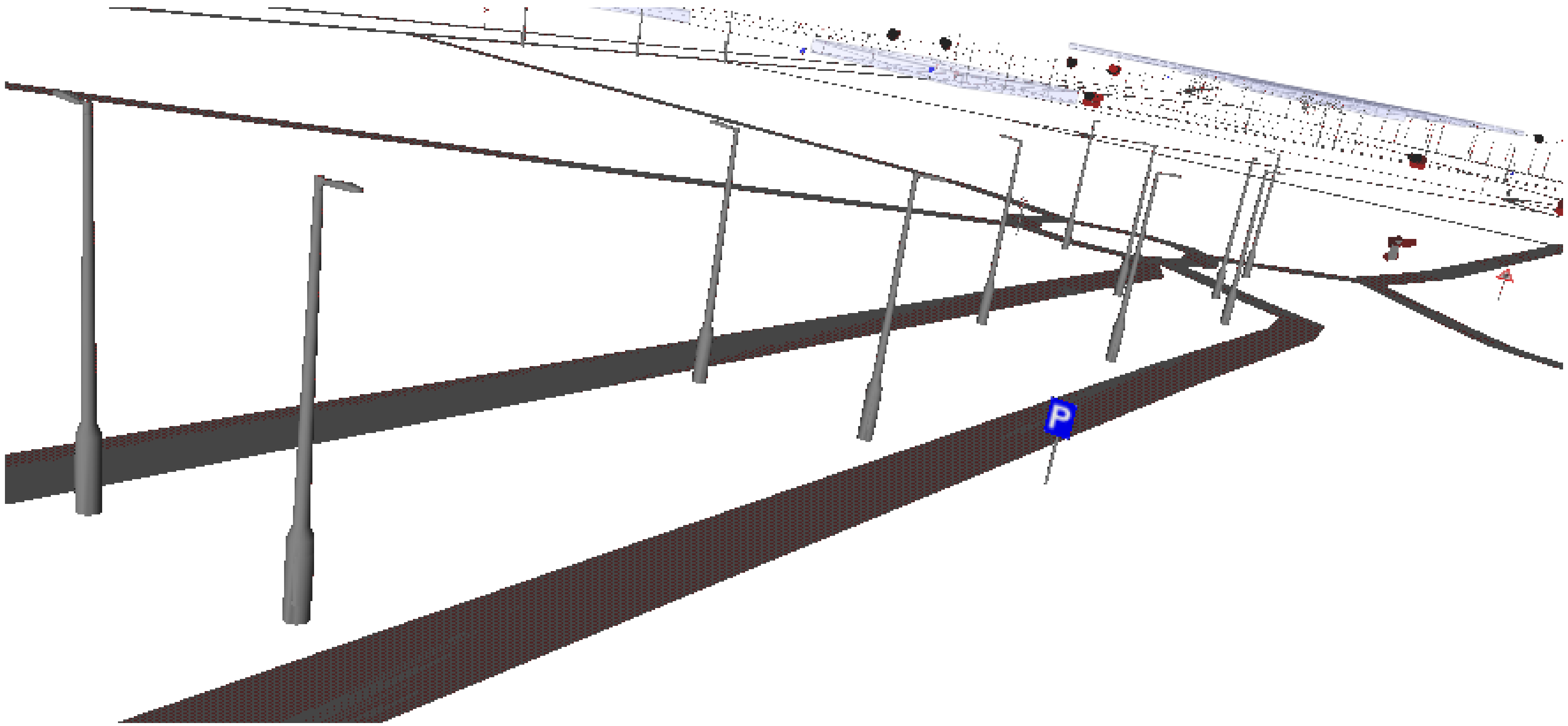






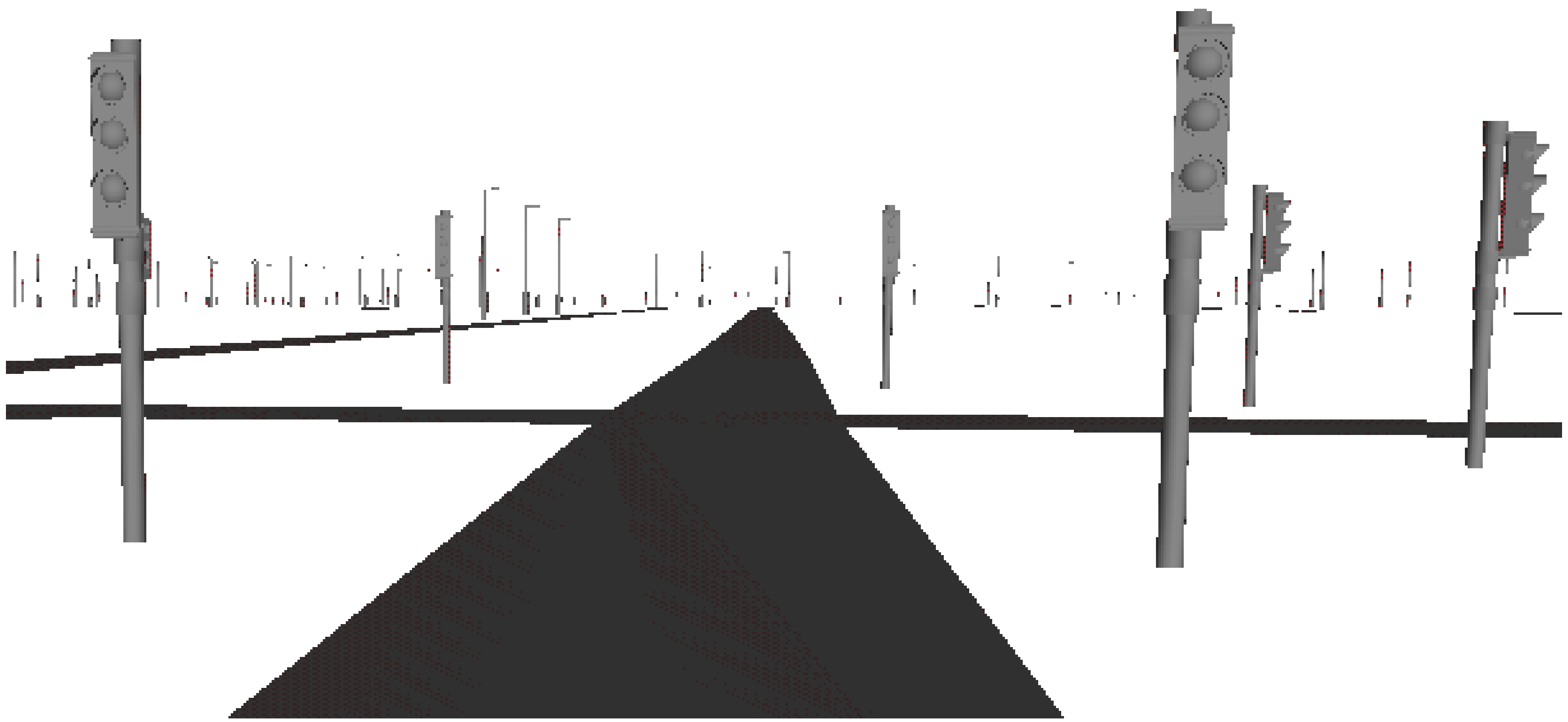
3D Modeling

Lamppost



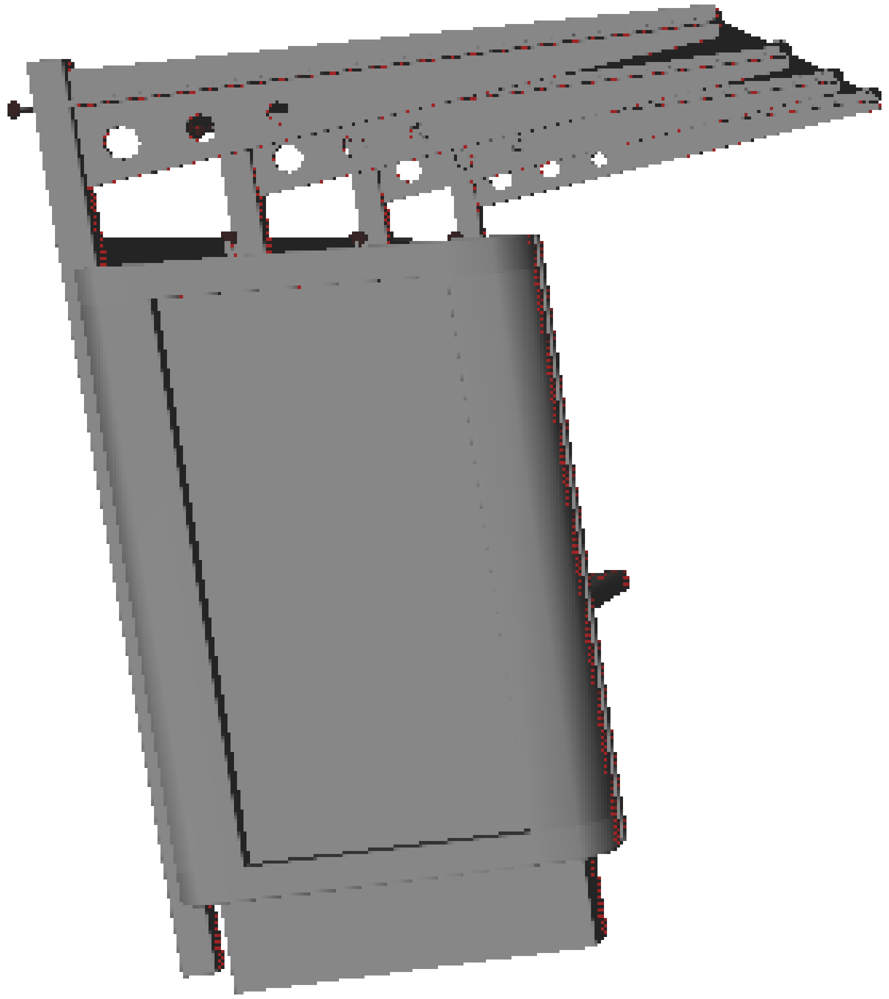
3D Modeling

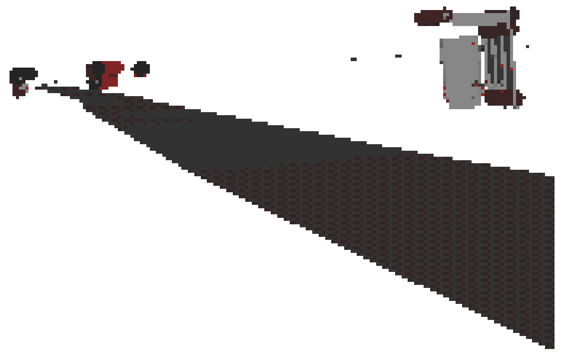
Traffic lights



3D Modeling

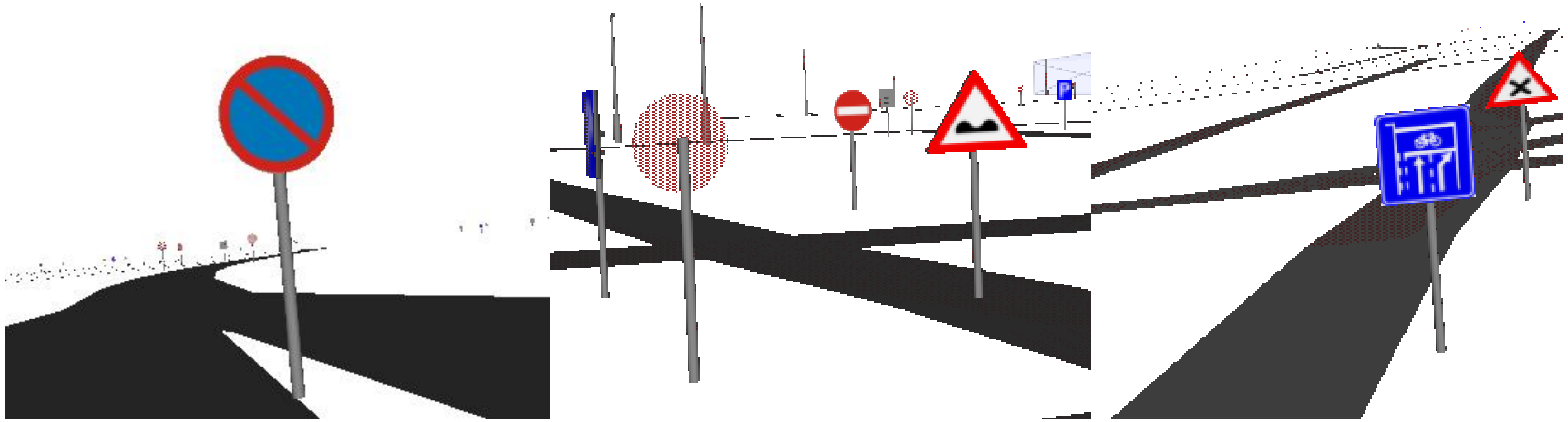
Bus station





3D Modeling

Traffic signs



Discussion

Comparison between Image-based approach and Camera LiDAR fusion Approach

	Imagery-based Approach	Camera-LiDAR Approach
Detection rate	Very High detection rate	Prone to omission and requiring more careful processing
Accuracy	0.32 RMSE compared to PICC data	Centemetric, Depending on the accuracy of the LiDAR system
Classification quality	Accurate Due to the maturaty of the mbject detection and classification models	Accurate for the imagery-based classification, but struggle with noise for the FGR/ICP registration method using point cloud

Discussion

Comparison between Image-based approach and Camera LiDAR fusion Approach

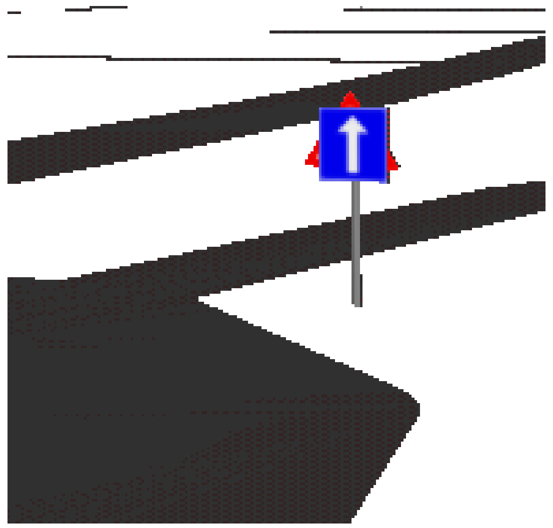
Traffic light Imagery based approach

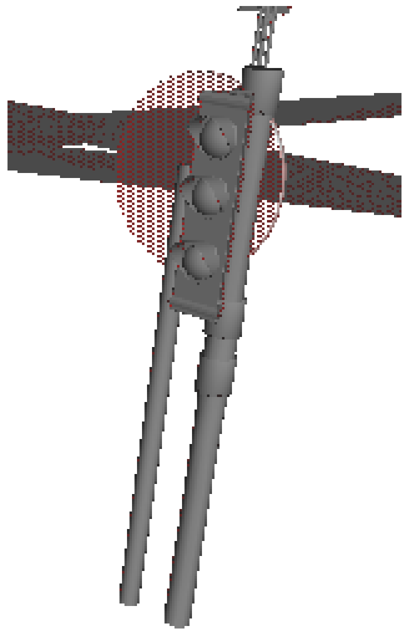
Traffic light Point Cloud approach

Discussion

Modeling limitations

There are more complex traffic signs, such as superimposed signs and merged traffic light and sign combinations, which require more intricate modeling. This issue could be addressed by performing spatial operations to cluster nearby objects, thereby diversifying the geometry templates for more complex models.





Conclusion

1. This study employed two approach for robust city furniture object detection, localization and modeling: an imagery-based approach and a camera-LiDAR fusion approach.

2. The imagery-based approach uses 360° images and trained YOLOv8 models for object detection, with Grounding DINO for fast label generation and a cascade detection/classification to classify traffic signs into 40 subclasses.

3. Object localization used photogrammetry and epipolar geometry, achieving high positional accuracy with an RMSE of 0.32 meters.

4. The camera-LiDAR fusion approach uses KPConv for 3D point cloud segmentation and LCC to separate instances, followed by classification using the already trained models.

5. Challenges included noise sensitivity in Fast Global Registration (FGR) during the camera-LiDAR fusion, suggesting a need for better outlier detection and potential use of advanced denoising techniques to improve classification accuracy.

6. Complex cases where traffic signs/lights are superimposed and merged need to be addressed in future studies.

8TH INTERNATIONAL ISPRS WORKSHOP

LOWCOST 3D - SENSORS, ALGORITHMS, APPLICATIONS

LIÈGE université
GeoScITY

Automatic Detection and 3D Modeling of City Furniture Objects using LiDAR and Imagery Mobile Mapping Data

Doi Hiba[1], Anass Yarroudh[2], Imane Jeddoub[2], Rafika Hajji[1], Roland Billen[2]

[1] College of Geomatic Sciences and Surveying Engineering, Hassan II IAV, Rabat 10101, Morocco
[2] GeoScITY, Geomatics Unit, UR SPHERES, University of Liège, Belgium

Thursday, 12/12/2024: 14:20 – 14:40

Università degli Studi di Brescia, Brescia, Italy