2025 IEEE Intelligent Vehicles Symposium (IV)
June 22-25, 2025. Cluj-Napoca, Romania

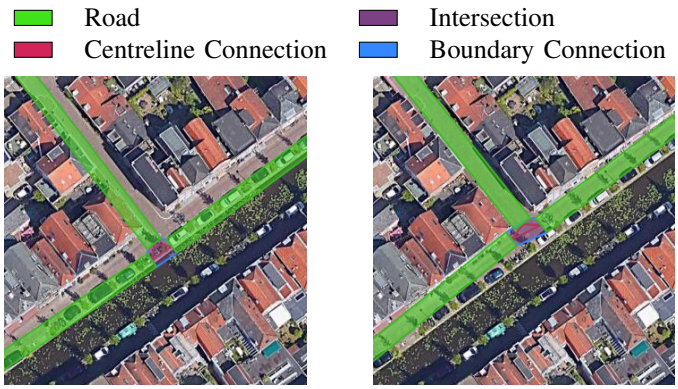
SAM-Maps: Road Map Generation for Automated Vehicles in Urban Areas

Matthijs P. van Andel*, Hidde J-H. Boekema*, and Dariu M. Gavrila
Intelligent Vehicles Group, Delft University of Technology

***Abstract*—Automated Vehicles (AVs) rely on up-to-date map information to inform trajectory prediction and planning modules, but these maps are expensive to obtain and update as they are usually annotated by humans. We propose SAM-Maps, a method for automatically generating road maps from aerial images of urban areas that takes advantage of the power of foundation models, requiring no human annotation or additional training to map unseen areas. This method extracts a coarse road graph from the images and then estimates the geometry of the roads from this graph.**

We evaluate our model on the challenging road layouts of the recent View-of-Delft Prediction dataset by comparing the maps generated using our model to the human-annotated maps, achieving an IoU of 33.3% with our automatic method and an IoU of 56.1% with some human corrections in our method. We also evaluate a trajectory prediction model on our maps to test whether they are sufficiently accurate for downstream tasks. The performance of this model using the map from our automatic method is 37.9% better on the minADE6 metric than not using map data as input. To the best of our knowledge, this is the first method that extracts both the drivable area and road connections of European urban areas from aerial images. The code will be publicly released for research purposes.

***Index Terms*—HD-Maps, Trajectory Prediction**



(a) OSM Map (b) SAM-Maps

Fig. 1: Intersection mapped with OSM [6] and SAM-Maps. Accurate OSM annotations are not available everywhere. Our method does not require human annotation and can extract the geometry and topology of urban roads from aerial images.

I. Introduction

It is essential for their widespread adoption that Automated Vehicles (AVs) can navigate urban areas without compromising the safety of surrounding road users. An understanding of the behaviour of surrounding agents is critical to this goal. This is especially important in urban areas, where AVs frequently interact with Vulnerable Road Users (VRUs) such as pedestrians and cyclists (e.g such as in [1], [2]). Trajectory prediction models help an AV achieve this understanding by estimating the future positions and/or intent of the agents around the vehicle. These models generally predict future trajectories from the observed trajectories of agents but can also use additional information about the agents or environment [3].

Current state-of-the-art trajectory prediction models rely heavily on road map data. Road maps contain information about the drivable area and connections of roads, making them a prior for where agents can go in the built environment. Recent trajectory prediction datasets, such as nuScenes [4], Argoverse 2 (AV2), Waymo Motion [5], and View-of-Delft Prediction (VoD-P) [1], provide high-definition (HD) road maps annotated by humans that additionally contain lane information. However, using human annotators is costly and can delay much-needed map updates when the static environment changes.

An alternative map source is OpenStreetMap [6] (OSM), which contains map annotations that can be used for AV tasks such as trajectory prediction [7]. These maps are primarily annotated by a community of volunteers, reducing annotation cost. However, this also makes them vulnerable to mistakes (see Figure 1) and even vandalism (see Figure 2). They are hence unsuited for safety-critical applications such as AVs.

Due to the high manual annotation cost of reliable maps, there is an active research community working on automatic road map generation from sensor data [8]. However, existing approaches either require data from expensive ground-based recording vehicles [9]–[15], do not estimate both the road geometry and topology [16]–[25], or may not generalise well to urban areas [26]–[28]. We propose a method that addresses these shortcomings. To the best of our knowledge, this is the first method that extracts both the drivable area and connections of roads in European urban areas from aerial images.

Our contributions are as follows:

1) We propose a method that uses RGB aerial images and foundation models to generate AV-suitable road maps (i.e. drivable area and road connections) of unseen urban areas without additional training or human annotation.

*Equal contribution

2025 IEEE Intelligent Vehicles Symposium (IV) | 979-8-3315-3803-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/IV64158.2025.11097443

Fig. 2: OSM maps can contain mistakes or even vandalism, such as this fictitious town drawn in farmland [29].

We show that these maps significantly improve trajectory predictions (compared to not using a map).

2) We investigate the impact of different map information on the coverage of roads in urban areas and on the performance of the state-of-the-art Wayformer [30] trajectory prediction model.
3) We release open-source software to aid geospatial graph and mapping operations[1].

II. Related Work

There is a large body of literature on generating road maps for Automated Vehicles (AVs) using the on-board sensors of a vehicle [9]–[15]. We focus on map generation methods that use aerial images in this section, as these methods do not suffer from the same restrictive mapping costs.

A. Road Graph Extraction

Many methods segment aerial images to extract the topology of the road network as a road graph (RG). [26] uses the DeepLabv3+ [31] to segment road areas and markings from aerial images. Similarly, AerialLaneNet [27] and [28] extract lane-level road maps by segmenting lane markings to find lane centrelines. However, these methods may not work for urban environments due to their dependence on lane markings, which are not always visible (see Figure 5 for an example). SAM-Road [16] instead estimates the centrelines directly, using the Segment Anything [32] model (SAM) to create a road graph from aerial images. This method only estimates the road topology and not its geometry, which is important in planning the trajectory of an AV. The DeepRoadMapper [17] and OrientationRefine [18] methods have the same shortcoming. Note also that these methods are trained on domain-specific aerial images and may not generalise to unseen areas.

Some methods instead opt for an iterative graph-growing approach to road topology estimation. These methods start from a known road point in the image and employ a search algorithm with a decision function to find connected roads, adding nodes and vertices to the graph when new roads are found [24], [25]. These methods do not estimate the road geometry either, however.

[1]https://github.com/tudelft-iv/sam-maps

B. Road Segmentation

Another essential component of road maps is the geometry of the roads. These can be segmented directly, or can be implicitly estimated by detecting road boundaries. The Topo-Boundary benchmark [19] contains $25,295$ large-scale RGB aerial images with 8 different labels for mapping tasks, such as road boundary and orientation detection. This benchmark paper proposes Enhanced-iCurb, a boundary detection method that has improved the training stability and convergence of iCurb [20], an imitation-learning-based approach for line-shaped object detection. Other road boundary detection methods include [21], a segmentation-based method that requires overhead LiDAR and camera data, csBoundary [22], which extracts boundary keypoints and adjacencies from aerial images to create a boundary graph, and Sat2Graph [23], which combine the advantages of segmentation-based and graph-based methods in a novel encoding scheme. Despite their applicability in road map generation and downstream AV tasks, boundary detection methods do not fully specify the road map, i.e. road geometry and topology. samgeo [33] can be used to segment regions in aerial images through user input, but cannot be used to create a road map fully automatically.

III. Proposed Method

Our method consists of three modules: **Road Graph Extraction (RGE)**, **Road Segmentation (RS)** and **Road Connection (RC)**. The **RGE module** extracts a coarse representation of road areas and connections that the **RS module** estimates the precise geometry of. The **RC module** reconnects the segmented roads and creates intersections. We describe these modules in detail below. An overview of our method is shown in Figure 3.

A. Road Graph Extraction (RGE)

A road graph (RG) is a global representation of the road network, consisting of **edges**, which represent road segments, and **nodes**, which denote connection points between the segments. We use two approaches for extracting the road graph: using OSM [6] or using SAM-Road [16].

1) Using OSM: OSM contains features relevant for making road graphs, specifically road centrelines, connectivity, and type (e.g., highway, cycle lane). Our OSM-based RGE module queries this information for the area to be mapped and formulates it as a road graph. The width is also annotated for some roads, enabling direct road segmentation by creating polygons from the centreline and width data.

2) Using SAM-Road: SAM-Road [16] generates road graphs from aerial images, eliminating the need for manual annotation. This method estimates the centrelines of the roads (as edges) and the connections between them (as nodes) but does not estimate the road width.

There may be mistakes in the road graph from either approach, such as missing roads or connections, or an incorrectly positioned road; all errors that can be easily fixed by a human annotator. We have therefore written a QGIS [34] plugin that makes it easy to move, add, and remove nodes and edges

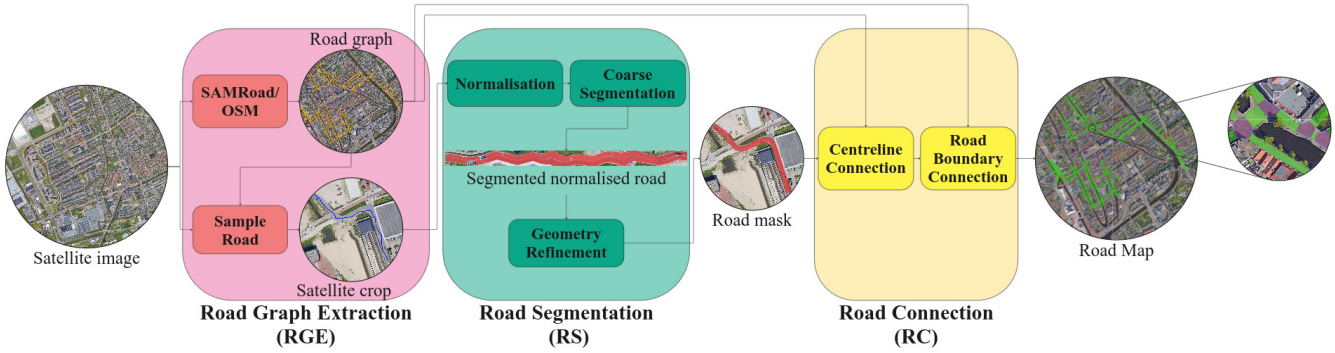


Fig. 3: Overview of our method. The **Road Graph Extraction (RGE)** module extracts a coarse representation of the roads and their connections. The **Road Segmentation (RS)** module estimates the precise geometry of the roads from this representation. The **Road Connection (RC)** module reconnects the segmented roads and creates intersections between them.

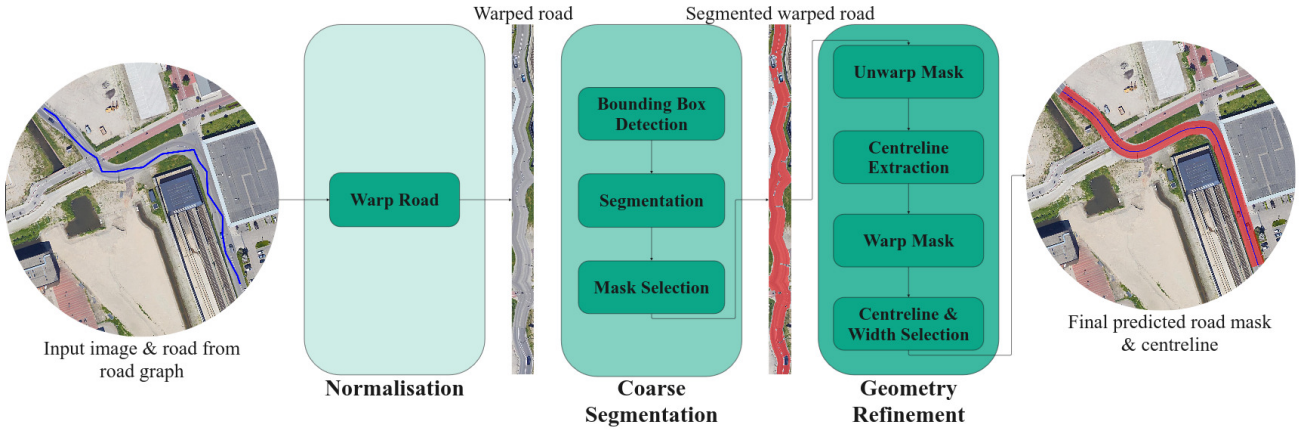


Fig. 4: Overview of the **Road Segmentation (RS)** module.

Fig. 5: Manual graph adjustment using our QGIS [34] plugin. Some of the nodes (green) are wrongly placed in a canal, but can be easily moved with one operation by a human annotator. The edges (blue) are automatically adjusted by the plugin.

without breaking the (road) graph. Figure 5 illustrates one such adjustment.

B. Road Segmentation (RS)

After extracting the coarse road graph, our method estimates the geometry of the roads in the graph. The RS module does this by the following steps (explained below): **normalisation**, **coarse segmentation**, and **geometry refinement**. Figure 4 illustrates these steps.

1) Normalisation: The first step normalises the format of the extracted road. Roads are straightened through piecewise **warping** using the nodes generated by the RGE module. The resulting image is then cropped to contain only the road surface from a initial estimate of the road width w_{init}. This step removes much of the variation in the image, making segmentation of the road easier.

2) Coarse Segmentation: The road surface is then segmented from the warped image. We use Grounding DINO [35], which detects objects based on a text prompt, to get **bounding box proposals** $\{\mathbf{b}_i\}_{i=1}^{N}$ around the road within the warped image. These are input to SAM [32] to generate a **segmentation mask** M_i^{road} for each proposal box $\mathbf{b}_i$.

Since the normalised image should represent a horizontal road, we fit a rectangular mask M^{rect} to the segmented mask in the **mask selection** step. The rectangular mask covers the entire length of the road, but its width is optimised to maximise the Intersection over Union (IoU) with the segmented mask. We select the segmented mask that achieves the highest IoU with its (optimised) rectangular mask. Masks are additionally required to adhere to constraints to be considered: 1) masks

must meet a minimum width requirement w_{min}, and 2) masks may not overlap excessively with tree, building, and water masks (generated using Grounded SAM [36]) as this indicates that they are a poor fit or represent the wrong class. This leads to the optimisation problem formulated in Equation (2).

$$\text{IoU}(A, B) = \frac{A \cap B}{A \cup B} \tag{1}$$

$$\begin{aligned} \arg\max_i \ \max_{a,b} \ & \text{IoU}(M^{\text{rect}}(a,b) \cap M_i^{\text{road}}) \\ \text{subject to:} \quad & |a-b| > w_{\text{min}}, \\ & \text{IoU}(M_i^{\text{road}}, M^{\text{tree}}) < c_{\text{tree}}, \\ & \text{IoU}(M_i^{\text{road}}, M^{\text{water}}) < c_{\text{water}}, \\ & \text{IoU}(M_i^{\text{road}}, M^{\text{building}}) < c_{\text{building}}. \end{aligned} \tag{2}$$

Here M_i^{road} represents a proposed segmented mask and M^{rect} is a rectangular mask with variable vertical bounds a and b. Finally, M^{tree}, M^{water} and M^{building} are the masks of the trees, water and buildings, respectively, and $c_{\text{tree}}, c_{\text{water}}$ and c_{building} are constants.

3) Geometry Refinement: The selected segmentation mask can be refined to fill in gaps in the mask and better estimate the width. First, the mask is warped back to the original coordinate system. A heatmap is then created from the mask using the SciPy [37] distance_transform_edt function, which is then max pooled to obtain the most likely centreline points. New centreline proposals are then created: cubic univariate splines with different smoothness parameters S and a line fit through the obtained points. Each centreline proposal is then used to warp the mask again, and then the mask selection step from the **RS** module is repeated for each centreline to select the best one; the road width is taken as $|a-b|$, e.g. the width of the best fitting rectangular box. This results in a complete road geometry i.e. road centreline and width.

C. Road Connection (RC)

To complete the road map, the segmented roads need to be connected to each other. The **RGE** module provides the connections between the roads but represents each connection as a node. However, the centrelines of the segmented roads and intersections between them may not align precisely with these nodes. To address this, the ends of the segmented roads from the **RS** module are trimmed, and a 'connection' line between the centrelines of the (trimmed) connected roads is added. An intersection polygon is additionally created by forming a convex hull around the endpoints of the (trimmed) connected roads to denote the drivable area at the intersection. Examples of intersection polygons are shown in purple in Figure 3. The generated connections ensure smooth and accurate transitions between roads, completing the road map.

IV. Experiments

We evaluate our map generation method on the accuracy of the maps compared to human-made annotations, as well as their usefulness in a downstream task. For these experiments, we select $w_{\text{init}} = 6$ m, $w_{\text{min}} = 4$ m, and $c_{\text{tree}} = c_{\text{water}} = c_{\text{building}} = 0.05$. For fitting splines, we use the SciPy [37] UniVariateSpline implementation with smoothness parameters $S = \{0.1, 0.2, 0.5, 1.0, 2.0, 3.5, 5.0, 10.0\}L$, where L is the length of the segmentation mask. These parameters were empirically determined.

TABLE I: Method nomenclature.

Category	Name	RGE	RS
Manual Annotation	OSM Map	OSM	OSM + Heuristic
	OSM-RG + A-RS	OSM	Auto-RS
	M-RG + A-RS	Manual (Man.)	Auto-RS
Manual Correction	SAM-Maps+	SAM-Road + Man.	Auto-RS
Automatic	SAM-Maps	SAM-Road	Auto-RS

Note that we do not train or fine-tune the foundation models for any of these experiments.

A. Road Map Coverage

To assess the fidelity of our generated road maps to the annotated maps provided with recent datasets, we evaluate the recall of the generated road map of the generated roads with respect to the annotated roads of the View-of-Delft Prediction (VoD-P) [1] dataset. This dataset provides challenging road layouts of the city of Delft. We also apply our method on Bratislava, another European city, to show its generalisability. We use (RGB) GeoTIFF images, sourced through the samgeo [33] interface, with a resolution of 8 cm/pixel, as input to our method.

We compare various mapping methods:

- *OSM Map*: a map generated from OSM data. Since OSM data may be available in the area to be mapped, we assess the relative quality of our approach to maps created from this data. We do this by estimating the drivable area from the centreline of annotated roads (that are labelled as drivable roads for vehicles) in combination with the annotated width. If the width is not annotated for a road, we set it to a default width (empirically selected as 4.5 m).
- *OSM-RG + A-RS*: our SAM-Maps RS and RC modules with the OSM road graph as input.
- *M-RG + A-RS*: our SAM-Maps RS and RC modules with a road graph drawn manually using the QGIS plugin.
- *SAM-Maps+*: our full SAM-Maps method, with the auto-generated road graph adjusted manually using the QGIS plugin to fix mistakes.
- *SAM-Maps*: our full SAM-Maps method without any human intervention.

An overview of these approaches is shown in Table I.

1) Quantitative results: For the quantitive evaluation we use the recall, precision, and intersection over union (IoU) metrics, defined as

$$\text{Recall} = \frac{M \cap M_{\text{GT}}}{M_{\text{GT}}}, \tag{3}$$

TABLE II: Map segmentation results and annotation time of various map sources on the VoD-P [1] dataset.

Name	Annotation Time (h) ↓	Recall (%) ↑	Precision (%) ↑	IoU (%) ↑
Annotated Map (GT)	~ 80	100	100	100
OSM Map	Many	64.0	**80.2**	55.3
OSM-RG + A-RS	Many	48.6	75.8	42.1
M-RG + A-RS	~ 2	**68.9**	74.8	**56.0**
SAM-Maps+	~ 0.5	**69.0**	**75.0**	**56.1**
SAM-Maps	0	**38.3**	**71.9**	**33.3**

$$\text{Precision} = \frac{M \cap M_{\text{GT}}}{M}, \tag{4}$$

$$\text{IoU} = \frac{M \cap M_{\text{GT}}}{M \cup M_{\text{GT}}}, \tag{5}$$

where M represents the generated mask of the road and M_{GT} represents the ground truth mask derived from the annotations from the VoD-P dataset. Note that the GT map annotations only span roads that were relevant to the scenarios recorded in VoD-P. To avoid penalising (valid) generated roads outside of the annotations, we make a buffer of 2 m around the annotated map and evaluate only generated masks in this region.

Table II presents the performance of the various methods on the metrics. We provide qualitative examples for SAM-Maps in Figures 1 and 6.

The highest IoU is achieved with SAM-Maps+. This method requires only manual correction of the road graph generated by SAM-Road. This is on par with using M-RG + A-RS. However, the latter requires full manual annotation of the road graph, which takes an annotator approximately 2 hours for the roads in the VoD-P dataset, compared to about 30 minutes of manual correction for SAM-Maps+. Both of these methods outperform the methods that use OSM. This is primarily due to roads having incorrect labels in OSM, resulting in roads missing from the map.

Finally, SAM-Maps performs worst on the IoU metric. This is mainly due to missing and misaligned (see Figure 5 for an example) roads in the graph generated by SAM-Road. This shows that some level of human correction is still important for optimal performance.

2) Failure Cases: There are failure cases where SAM-Maps generates a mask that does not fit the masks of the annotated maps. Figure 6 shows examples of some of these cases.

The first example shows a road for which the geometry is incorrectly estimated. The mask for the rightmost road suggests that it turns just before the intersection instead of continuing straight towards it. This could affect downstream tasks such as trajectory prediction and planning by biasing them towards following the non-existent turn. This example shows that SAM-Maps could benefit from better handling of intersections using the incoming and outgoing lanes in the road graph, e.g. smoothing the transitions between nodes.

The second example exposes another limitation of the model: its inability to predict individual lanes. In this case, the number of lanes decreases from three to two, but SAM-Maps fails to capture this change. It instead estimates a best fit for the road that combines the width of the three lanes, extending into the narrowed section. Evaluating the individual lanes could therefore improve the performance.

Fig. 6: Qualitative examples of failure cases of SAM-Maps, showing the annotated (left) and generated (right) maps.

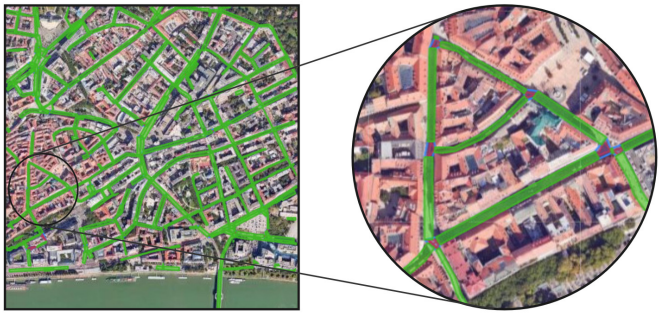


Fig. 7: Output of SAM-Maps on the city of Bratislava. No additional training or annotation was needed to create this map.

3) Generalisability: We applied SAM-Maps to aerial images of Bratislava without any additional training or human intervention. There are no annotations available for this data, making a quantitative analysis difficult, but the qualitative result shown in Figure 7 gives an indication of the generalisability of the method. Our approach is able to map large areas of the city correctly without any annotation effort.

B. Ablation Study

We conduct an ablation study to determine the value of the key components in our method. We use the SAM-Maps+ method (with corrected road graph) because it achieves the best results out of the variations that use all components of our method. The results presented in Table III highlight the importance of each component of our method.

The modules in SAM-Maps can be further broken down into key operations, including **mask segmentation**, **normalisation**, **bounding box proposal**, **mask selection**, and **geometry refinement**. We systematically ablate these components to assess their impact. Since mask selection is inherently tied to

TABLE III: Ablation study results using SAM-Maps+.

Mask Seg.	Normal.	Bbox Prop.	Geom. Ref.	Recall (%) ↑	Precision (%) ↑	IoU (%) ↑
☐	☐	☐	☐	64.8	71.2	51.3
☑	☐	☐	☐	35.3	60.4	28.7
☑	☑	☐	☑	56.7	72.2	46.5
☑	☑	☑	☐	**70.0**	74.0	56.0
☑	☑	☑	☑	69.0	**75.0**	**56.1**

Fig. 8: Example of the effect of geometry refinement.

the normalisation and bounding box proposal steps, we ablate these steps together.

1) Mask Segmentation: When the segmentation step is ablated, the method cannot determine road width. The edges generated by the RGE module are then taken as centrelines, and a fixed width (of 4.5 m) is assigned to all roads. This method, therefore, depends on the quality of the graph alignment and fails to account for road width variations. As shown in the top row of Table III, this results in a IoU drop of 4.8%, underscoring the importance of segmentation in handling diverse road geometries.

2) Normalisation: The normalisation step reduces background noise and standardises the roads in the images by warping them. Without warping, the segmentation masks often capture irrelevant objects, such as buildings or vegetation, leading to noisy or incorrect results. This causes the IoU of the method to halve, emphasising the need for a normalisation step.

3) Bounding Box Proposal: Although the normalisation step standardises the image of the road, uncertainty in the road width and road graph alignment remains. Without the bounding box proposals, the entire normalised image is input to SAM, which often segments irrelevant objects (e.g. trees or cars) as part of the road surface. This again leads to a significant drop in performance.

4) Geometry Refinement: The geometry refinement module extracts and selects new road centrelines and widths from the selected segmentation mask. This step has little impact on performance; SAM-Maps+ even performs slightly better on the recall metric without this module. This can be explained by how the width estimation algorithm works: it is based on the best fit of a rectangular box on the warped mask of the road. If the nodes in the road graph are inaccurate, the centreline and segmented mask may snake across the road. This, in turn, leads to wider rectangular fits, leading to overestimation of the road width. Hence the method can perform similarly (or even better) on the recall metric without geometry refinement, but the resulting road map can adversely affect downstream tasks such as trajectory prediction. An example of a centreline that can cause this is shown in Figure 8. This figure also illustrates the smoothed road centreline that the geometry refinement produces.

TABLE IV: Road boundary detection results on the VoD-P [1] dataset.

Method Name	Recall (%) ↑		
	$\tau = 2$	$\tau = 5$	$\tau = 10$
OrientationRefine [18]	0.328	0.832	1.79
Enhanced-iCurb [19]	1.99	5.23	10.4
SAM-Maps	**6.29**	**16.2**	**28.9**
SAM-Maps+	**12.3**	**35.1**	**66.7**

TABLE V: Trajectory prediction results of Wayformer [30] on VoD-P [1] with various map information.

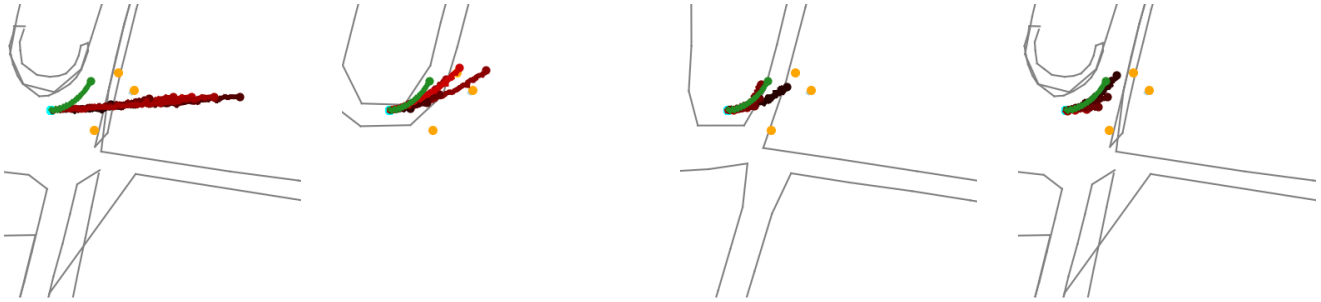
Map Information	Automatic	minADE6 (m) ↓	MissRate6 ↓
SAM-Maps+	☐	1.80	0.38
OSM Map	☐	1.70	0.29
Annotated Map	☐	**1.23**	**0.28**
No Map	☑	2.98	0.55
SAM-Maps	☑	**1.85**	**0.40**

C. Topological Road Boundary Detection

We can easily extract road boundaries from the map produced by SAM-Maps, enabling comparison with road boundary detection methods from the literature. We choose the OrientationRefine [18] and Enhanced-iCurb [19] methods as baselines to compare to because they are the top-performing methods on the Topo-boundary [19] benchmark that additionally have a public implementation. These methods need RGB+NIR images with a resolution of approximately 15 cm/pixel, so we source openly-available RGB and NIR images of Delft from PDOK[2] and upsample these from 25 cm/pixel to 15 cm/pixel. Note that we do not train any of the underlying models for the city Delft since they should be able to generalise to new areas. Following [19], we evaluate the per-pixel recall of the boundaries where the predicted boundary pixels may lie within a threshold τ of the ground truth boundary pixels. In Table IV, we report results for $\tau = \{2, 5, 10\}$ pixels.

The performance of the baselines is significantly worse than that of SAM-Maps and SAM-Maps+, with the best baseline, Enhanced-iCurb, scoring only 10.4% on the recall metric with $\tau = 10$ against a score of 66.7% for SAM-Maps+. This shows that the baselines cannot generalise well to a European urban area such as Delft.

[2]https://www.pdok.nl



(a) No Map (b) OSM Map (c) SAM-Maps (d) Annotated Map

Fig. 9: Qualitative trajectory prediction results on the VoD-P dataset with the Wayformer model and different map inputs. Predictions are shown in red, ground truth future in green, and other agents in orange. Annotated map shown for the 'No Map' setup for clarity. The model using our SAM-Maps map correctly predicts the turn, but the model without map does not.

D. Trajectory Prediction

We further test our maps by using them as input data in the trajectory prediction task. The goal in this task is to estimate the future poses $\{\mathbf{x}_{1:T_f}\}_A$ of a set of agents A from their observed past states $\{\mathbf{x}_{-T_h:0}\}_A$ and map information $\mathcal{M}$.

We select the Wayformer [30] trajectory prediction model for our experiments as it is a recent high-performing model on the Waymo Motion [5] benchmark and has an architecture that is easily modified to work without map data. We use the UniTraj[3] [38] open-source implementation of this model.

We evaluate the model on the VoD-P dataset to test the generated maps for this dataset. We compare the performance of the model using a map from our SAM-Maps approach to its semi-automatic variant SAM-Maps+ and the OSM map to assess their quality. Since VoD-P has a relatively small number of scenarios compared to other public datasets, we pre-train the model on the nuScenes dataset for 150 epochs and fine-tune the best model (on the validation data) on the VoD-P dataset for 300 epochs. We convert both datasets to the ScenarioNet [39] format to homogenise the data, and make training and evaluation scenarios with a short history of 0.5s and a long future (prediction) horizon of 6s to encourage the model to use the map data, when available, in its predictions.

Table V shows the results on the minimum average displacement (minADE) and miss rate (MissRate) metrics for 6 predictions. The performance of the model without map information is poor on both metrics, with all map-based models outperforming it by a significant margin. None of the automatically-generated maps leads to the same performance as the annotated map. Notably, the model performs 38.2% worse on the minADE6 metric with the (annotated) OSM map than with the annotated VoD-P map. The drop in performance when using the SAM-Maps+ and SAM-Maps maps instead of the annotated OSM map is less than 10% on minADE6 and around 0.10 points on MissRate6. SAM-Maps+ only slightly outperforms SAM-Maps, at the cost of some human annotation effort.

[3]https://github.com/vita-epfl/UniTraj

Figure 9 shows qualitative results for some of the results in Table V. The prediction model is unable to infer that the vehicle will turn without map input in the example shown, whereas it correctly estimates the turn with the SAM-Maps map. This map also has better coverage than the OSM map, which does not contain some of the roads. These results confirm the usefulness of the maps of the SAM-Maps method for the trajectory prediction task.

V. Conclusion

We presented a method for generating road maps containing the drivable area and road connections of unseen urban areas from aerial images without needing human annotation, significantly cutting annotation cost and time. These maps can, however, be easily edited by humans to fix errors made in the automatic pipeline through software that we developed. We evaluated the maps created by this method by comparing them to human-annotated maps of the European city of Delft, and found that the auto-generated map has an IoU of 33.3% with the annotated map, rising to 56.1% using our semi-auto-generated map. We further tested these maps for the AV task of trajectory prediction on the urban View-of-Delft Prediction [1] dataset, and found that using our auto-generated map led to a performance improvement of 37.9% (and over 1.0 m) on the minADE6 metric compared to using no map as input to the state-of-the-art Wayformer model [30]. Future work includes segmenting the lanes of a road individually and improving the estimation of the geometry of intersections.

Acknowledgement

This work was supported by the Dutch Research Council (NWO) under project number P16-25 (Efficient Deep Learning - Mobile Robotics) and the European Union (EU) under Grant 101069614 (EVENTS project). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the NWO, EU or European Commission. Neither can be held responsible for them.

REFERENCES

[1] H. J.-H. Boekema, B. K. Martens, J. F. Kooij, and D. M. Gavrila, "Multi-class trajectory prediction in urban traffic using the view-of-delft prediction dataset," *IEEE Rob. and Aut. Lett.*, vol. 9, no. 5, pp. 4806–4813, 2024.

[2] S. Krebs, M. Braun, and D. M. Gavrila, "Eurocity persons 2.0: A large and diverse dataset of persons in traffic," *IEEE Trans. on Patt. Analysis and Machine Int.*, vol. 46, no. 12, pp. 10 929–10 943, 2024.

[3] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The Intl. Journal of Rob. Research*, vol. 39, no. 8, pp. 895–935, 2020.

[4] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *Proc. of the IEEE/CVF Conf. on Comp. Vis. and Patt. Rec.*, 2020, pp. 11 621–11 631.

[5] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou *et al.*, "Large scale interactive motion forecasting for autonomous driving: The Waymo Open Motion dataset," in *Proc. of the IEEE/CVF Intl. Conf. on Comp. Vis.*, 2021, pp. 9710–9719.

[6] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2017.

[7] J.-Y. Liao, P. Doshi, Z. Zhang, D. Paz, and H. Christensen, "OSM vs HD maps: Map representations for trajectory prediction," in *2024 IEEE/RSJ Intl. Conf. on Int. Robots and Systems (IROS)*, 2024, pp. 9990–9996.

[8] Z. Bao, S. Hossain, H. Lang, and X. Lin, "High-definition map generation technologies for autonomous driving," *arXiv preprint arXiv:2206.05400*, 2022.

[9] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "HDMapNet: An online hd map construction and evaluation framework," in *2022 Intl. Conf. on Rob. and Aut. (ICRA)*. IEEE, 2022, pp. 4628–4634.

[10] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, and C. Huang, "MapTR: Structured modeling and learning for online vectorized HD map construction," *arXiv preprint arXiv:2208.14437*, 2022.

[11] Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao, "VectorMapNet: End-to-end vectorized HD map learning," in *Intl. Conf. on Machine Learning*. PMLR, 2023, pp. 22 352–22 369.

[12] W. Ding, L. Qiao, X. Qiu, and C. Zhang, "Pivotnet: Vectorized pivot learning for end-to-end hd map construction," in *Proc. of the IEEE/CVF Intl. Conf. on Comp. Vis.*, 2023, pp. 3672–3682.

[13] Y. Cai, W. Dong, Z. Liu, H. Wang, and L. Chen, "HoMap: End-to-end vectorized hd map construction with high-order modeling," *IEEE Trans. on Int. Vehicles*, 2024.

[14] K. Tang, X. Cao, Z. Cao, T. Zhou, E. Li, A. Liu, S. Zou, C. Liu, S. Mei, E. Sizikova *et al.*, "THMA: Tencent Hd map AI system for creating HD map annotations," in *Proc. of the AAAI Conf. on Artificial Int.*, vol. 37, no. 13, 2023, pp. 15 585–15 593.

[15] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun, "HD maps: Fine-grained road segmentation by parsing ground and aerial images," in *Proc. of the IEEE Conf. on Comp. Vis. and Patt. Rec.*, 2016, pp. 3611–3619.

[16] C. Hetang, H. Xue, C. Le, T. Yue, W. Wang, and Y. He, "Segment Anything model for road network graph extraction," in *Proc. of the IEEE/CVF Conf. on Comp. Vis. and Patt. Rec.*, 2024, pp. 2556–2566.

[17] G. Máttyus, W. Luo, and R. Urtasun, "DeepRoadMapper: Extracting road topology from aerial images," in *Proc. of the IEEE Intl. Conf. on Comp. Vis.*, 2017, pp. 3438–3446.

[18] A. Batra, S. Singh, G. Pang, S. Basu, C. Jawahar, and M. Paluri, "Improved road connectivity by joint learning of orientation and segmentation," in *Proc. of the IEEE/CVF Conf. on Comp. Vis. and Patt. Rec.*, 2019, pp. 10 385–10 393.

[19] Z. Xu, Y. Sun, and M. Liu, "Topo-boundary: A benchmark dataset on topological road-boundary detection using aerial images for autonomous driving," *IEEE Rob. and Aut. Lett.*, vol. 6, no. 4, pp. 7248–7255, 2021.

[20] ——, "iCurb: Imitation learning-based detection of road curbs using aerial images for autonomous driving," *IEEE Rob. and Aut. Lett.*, vol. 6, no. 2, pp. 1097–1104, 2021.

[21] J. Liang, N. Homayounfar, W.-C. Ma, S. Wang, and R. Urtasun, "Convolutional recurrent network for road boundary extraction," in *Proc. of the IEEE/CVF Conf. on Comp. Vis. and Patt. Rec.*, 2019, pp. 9512–9521.

[22] Z. Xu, Y. Liu, L. Gan, X. Hu, Y. Sun, M. Liu, and L. Wang, "csBoundary: City-scale road-boundary detection in aerial images for high-definition maps," *IEEE Rob. and Aut. Lett.*, vol. 7, no. 2, pp. 5063–5070, 2022.

[23] S. He, F. Bastani, S. Jagwani, M. Alizadeh, H. Balakrishnan, S. Chawla, M. M. Elshrif, S. Madden, and M. A. Sadeghi, "Sat2Graph: Road graph extraction through graph-tensor encoding," in *Comp. Vis.–ECCV 2020: 16th European Conf., Glasgow, UK, August 23–28, 2020, Proc., Part XXIV 16*. Springer, 2020, pp. 51–67.

[24] F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, S. Madden, and D. DeWitt, "RoadTracer: Automatic extraction of road networks from aerial images," in *Proc. of the IEEE Conf. on Comp. Vis. and Patt. Rec.*, 2018, pp. 4720–4728.

[25] Y.-Q. Tan, S.-H. Gao, X.-Y. Li, M.-M. Cheng, and B. Ren, "VecRoad: Point-based iterative graph exploration for road graphs extraction," in *Proc. of the IEEE/CVF Conf. on Comp. Vis. and Patt. Rec.*, 2020, pp. 8910–8918.

[26] G.-W. Chen and H.-Y. Lai, "Extracting high definition map information from aerial images," in *Workshop Proc. of the 51st Intl. Conf. on Parallel Processing*, 2022, pp. 1–5.

[27] J. Yao, X. Pan, T. Wu, and X. Zhang, "Building lane-level maps from aerial images," in *ICASSP 2024-2024 IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 3890–3894.

[28] S. He and H. Balakrishnan, "Lane-level street map extraction from aerial imagery," in *Proc. of the IEEE/CVF Winter Conf. on Applications of Comp. Vis.*, 2022, pp. 2080–2089.

[29] OpenStreetMap Wiki, "Vandalism — OpenStreetMap Wiki," 2024, [Accessed: 2025-03-25]. [Online]. Available: https://wiki.openstreetmap.org/w/index.php?title=Vandalism&oldid=2758449

[30] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion forecasting via simple & efficient attention networks," in *2023 IEEE Intl. Conf. on Rob. and Aut. (ICRA)*. IEEE, 2023, pp. 2980–2987.

[31] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. on Patt. Analysis and Machine Int.*, vol. 40, no. 4, pp. 834–848, 2017.

[32] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment Anything," in *Proc. of the IEEE/CVF Intl. Conf. on Comp. Vis.*, 2023, pp. 4015–4026.

[33] Q. Wu and L. P. Osco, "samgeo: A python package for segmenting geospatial data with the Segment Anything model (SAM)," *Journal of Open Source Software*, vol. 8, no. 89, p. 5663, 2023.

[34] QGIS Development Team, *QGIS Geographic Inf. System*, QGIS Association, 2025. [Online]. Available: https://www.qgis.org

[35] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, "Grounding Dino: Marrying dino with grounded pre-training for open-set object detection," in *European Conf. on Comp. Vis.* Springer, 2025, pp. 38–55.

[36] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan *et al.*, "Grounded SAM: Assembling open-world models for diverse visual tasks," *arXiv preprint arXiv:2401.14159*, 2024.

[37] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[38] L. Feng, M. Bahari, K. M. B. Amor, É. Zablocki, M. Cord, and A. Alahi, "Unitraj: A unified framework for scalable vehicle trajectory prediction," in *European Conf. on Comp. Vis.* Springer, 2025, pp. 106–123.

[39] Q. Li, Z. M. Peng, L. Feng, Z. Liu, C. Duan, W. Mo, and B. Zhou, "Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling," *Advances in Neural Inf. Processing Systems*, vol. 36, 2024.