Patched Line Segment Learning for Vector Road Mapping

Jiakun Xu[1], Bowen Xu[1], Gui-Song Xia[1], Liang Dong[2], Nan Xue[*1,3]

[1]School of Computer Science, Wuhan University
[2]Google Inc.
[3]Ant Group
{jiakun.xu, xbw, guisong.xia}@whu.edu.cn, lidong@google.com, xuenan@ieee.org

Abstract

This paper presents a novel approach to computing vector road maps from satellite remotely sensed images, building upon a well-defined Patched Line Segment (PaLiS) representation for road graphs that holds geometric significance. Unlike prevailing methods that derive road vector representations from satellite images using binary masks or keypoints, our method employs line segments. These segments not only convey road locations but also capture their orientations, making them a robust choice for representation. More precisely, given an input image, we divide it into non-overlapping patches and predict a suitable line segment within each patch. This strategy enables us to capture spatial and structural cues from these patch-based line segments, simplifying the process of constructing the road network graph without the necessity of additional neural networks for connectivity. In our experiments, we demonstrate how an effective representation of a road graph significantly enhances the performance of vector road mapping on established benchmarks, without requiring extensive modifications to the neural network architecture. Furthermore, our method achieves state-of-the-art performance with just 6 GPU hours of training, leading to a substantial 32-fold reduction in training costs in terms of GPU hours.

1 Introduction

By "vector road mapping", it refers to a process of converting the road features presented in satellite-borne remote sensing images into vector-based and symbolic graph representations, which is also known as *road graph extraction* or *road network extraction* within the community of remote sensing and plays a fundamental role in numerous downstream tasks including navigation (Zhang et al. 2021; Cai et al. 2023), urban planning (Shi et al. 2019; Xu et al. 2023a), and autonomous driving (Xu, Sun, and Liu 2021; He and Balakrishnan 2022; Büchner et al. 2023).

The state-of-the-art methods for vector road mapping primarily rely on the strong representation capabilities of deep neural networks. These approaches formulate the problem as a supervised learning task, utilizing paired satellite images and annotated road graphs that use vertices and edges to depict the line and curve structures of roads. As the input

---

(a) Input image (b) Ground truth (c) Dense GT

(d) Road graph learned from Keypoints (Xu et al. 2023b)
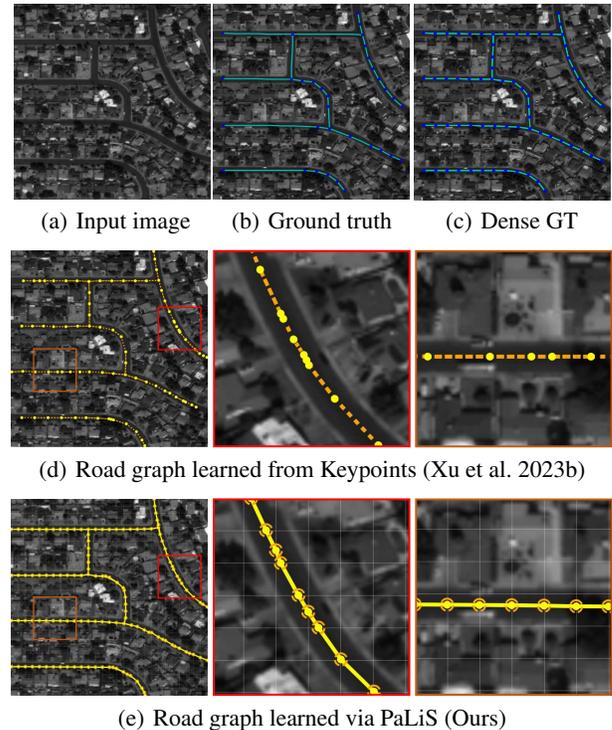

(e) Road graph learned via PaLiS (Ours)

Figure 1: Illustration of graphs constructed by different representations. The predicted representations (keypoints and line segments) are denoted in yellow marks and the connectivities are denoted in orange marks.

images are in pixel form, it becomes crucial to establish an appropriate representation for facilitating the learning from the pixels of satellite images to the vector representation of roads. In the state-of-the-art methods (Batra et al. 2019; He et al. 2020; Xu et al. 2023b), the "appropriate representation" of vector road annotations were initially come down to mask-based representation (*i.e.*, road masks) and were then upgraded to the keypoint-based graph representation as the main representation in the pursuit of end-to-end learning.

While keypoint-based graph representations have demonstrated remarkable performance, many of these methods encounter a significant drawback: the substantial training cost

involved. For instance, RNGDet++ model (Xu et al. 2023b) requires approximately 192 GPU hours to train on a dataset of moderate size with thousands of images. This high training cost can be attributed to the prevalent oversampling strategy used to define the "keypoints" in the original annotations (depicted in Fig. 1(b)). This strategy involves densely sampling numerous points along each road, as shown in Fig. 1(c), lacking invariance to commonly employed image transformations used for data augmentation, such as random cropping and image translation, and eventually results in ambiguity during the learning process. As a consequence, methods employing keypoint-based graph representations must grapple with inherent representation ambiguity, requiring a greater number of training iterations. Such a prolonged training process often entails cluttered patterns in the keypoint detection outcomes, as illustrated by the enclosed regions in Fig. 1(d). Furthermore, the keypoint-based representations have to leverage additional modules to learn the connectives for the learned keypoints on the fly to accomplish the task of vector road mapping.

In this paper, we devote ourselves to finding a better representation of vector road annotations, to eliminate the ambiguity in the existing keypoint-based graph representations, for the sake of efficient learning during training and top-performing mapping results in the testing phase. Our study is motivated by the recently-proposed PaRK-Detect (Xie et al. 2023) that defines *patched keypoints*, in which each small patch (*e.g.*, 16×16) will have at most one keypoint for learning. Because the local patches are uniformly distributed over the image grids, such a definition largely eliminates the ambiguity for learning. However, since the keypoints are unary primitives that do not explicitly define the spatial relationships, PaRK-Detect (Xie et al. 2023) only obtained comparable performance in testing. Motivated by this, we are interested in presenting a patched representation to take its ambiguity-free merits while retaining the spatial context for facilitating the final vector road mapping.

Our work is inspired by an observation that *the spatial and geometric information of roads in local patches can be well represented by line segments instead of keypoints*. Based on this, we present a novel PaLiS (Patched Line Segment) representation to depict the annotated road graphs in a geometrically-meaningful way while enjoying the ambiguity-free merits of patch-based representation. By dividing the grid of input images into a set of local (*e.g.*, 8×8) patches, most of the local patches that contain a fragment of road path can uniquely define the only local line segment. To preserve the rich structural information of the local line segments, we use the closed-form $xy - xy$ representation for the two endpoints of a line segment, which facilitates the computation of patch adjacency in a geometrically-meaningful way. As shown in Fig. 1(e), our proposed PaLiS representation could handle a variety of road graph patterns in a unified representation. With the proposed PaLiS representation, we find out that our PaLiS representation can be reliably learned via the rasterized road masks as supervision in differentiable rasterization, largely alleviating the need for vectorized road graph annotations.

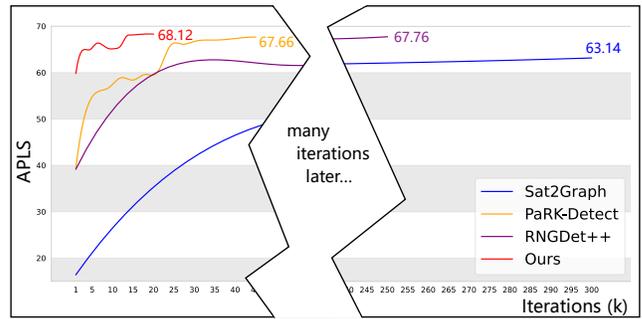In the experiments, we demonstrate that our proposed

Figure 2: Convergence curves on City-Scale dataset.

PaLiS representation clearly set new state-of-the-art performance on two public benchmarks, *i.e.*, the City-Scale (He et al. 2020) and SpaceNet (Van Etten, Lindenbaum, and Bacastow 2018), without paying any extra efforts on the network design. Except for the competitive performance on these two benchmarks, our method only requires 6 GPU hours for the training, significantly reducing the training cost by 32 times for the prior art, RNGDet++ (Xu et al. 2023b). As shown in Fig. 2, for the performance evaluation by training iterations on the City-Scale dataset, our proposed method wins after the first 1K iterations by significant margins and converges to the S.O.T.A. performance after 20K iterations of training.

In summary, our paper made the following contributions:

- We propose a novel representation of road graphs, the patched line segment representation, which facilitates the learning of road graphs with the best efficacy in both the training and testing phases.

- Based on our patched line segment representation, we present a graph construction strategy for the task of vector road mapping, which takes advantage of the geometric nature of our representation to produce vector graphs without using any additional neural networks for the learning of connectives between keypoints.

- Our proposed patched line segment representation is learnable and compatible with the mask-based representation by leveraging a differentiable soft rasterizer, which helps to learn the patched line segments efficiently without introducing additional vector labels.

2   Related Works

Road Graph Representations.   There have been plenty of studies for vector road mapping, mainly relying on either the rasterized road map or the keypoint/vertex-based graph representations, and derived two categories, the segmentation-based (Máttyus, Luo, and Urtasun 2017; Zhou, Zhang, and Wu 2018; Mei et al. 2021; Wang et al. 2023; Batra et al. 2019; Cheng et al. 2021) and the keypoint-based approaches (He et al. 2020; He, Garg, and Chowdhury 2022; Shit et al. 2022; Yang et al. 2023; Xie et al. 2023). Regarding the popularity of end-to-end learning for better performance, the state-of-the-art approaches (He, Garg, and Chowdhury 2022; Xu et al. 2023b) mainly learn keypoints (*i.e.*, graph

(a) The Road Ground Truth   (b) PaLiS Representation

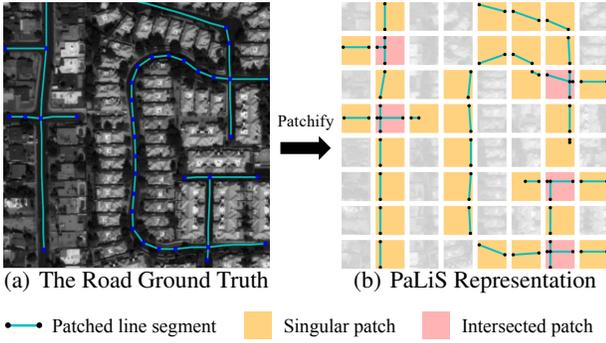•—•—• Patched line segment   ▨ Singular patch   ▨ Intersected patch

Figure 3: An illustrative figure for the proposed Patched Line Segment (PaLiS) representation. Larger patch size is applied for better illustration.

vertices) and the connectivity between vertices while using the rasterized road masks/maps as the additional supervision signals to enhance the feature representation ability of ConvNets. Except for the representation ambiguity issue discussed in Sec. 1 for prolonged learning schedule, these representations mainly focus on point primitives instead of the line structure of road graphs, thus usually requiring additional design to learn or infer the connectivity between points/pixels. Regarding the above issues, we present a novel line-segment-based representation that defines the road graphs in the local image patches while characterizing the structural information of roads using line segments. We show that our well-defined and geometrically-meaningful representation largely facilitates the learning process of vector road mapping with the best efficacy.

Line Segment Learning and Differentiable Rasterization. There has been a vast body of literature studying the line segments from both computer vision (CV) and graphics (CG) communities. On one hand, many works study the problem of line segment detection (Xue et al. 2019, 2020, 2021, 2022), which is similar to vector road mapping but mainly focuses on the line segment itself instead of the road graphs. On another hand, some CG researchers study the differentiable vector graphics rasterization/rendering (Li et al. 2020; Xie et al. 2014), in which they aim at using graphic primitives such as points, lines, and curves to represent rasterized digital images. The differentiable rasterization techniques were also applicable to the polygonal shape representation with end-to-end learning in instance segmentation (Lazarow, Xu, and Tu 2022) and polygonal building extraction (Zorzi et al. 2022). Our study is inspired by all these studies, but we pay more attention on the well-posedness of the primitive definition for the complicated road graphs/networks. By thinking of local patches, we eventually derive our novel PaLiS representation and set new state-of-the-art performance for the task of vector road mapping.

3 PaLiS Representation of Road Graphs

In this section, we elaborate on the proposed PaLiS representation of road graphs. Denoted by the input satellite image $\mathbf{I} \in \mathbb{R}^{3 \times H \times W}$ and the corresponding road graph an-

notation $\mathcal{R} = \{\Gamma_i(t) \in \mathbb{R}^2 | t \in [0, 1]\}$, where $\Gamma_i(t)$ is a parameterized 2D curve/line, $\Gamma_i(0)$ and $\Gamma_i(1)$ respectively represent the two endpoints of the parameterized curve. We use the local $p \times p$ patches to patch-wisely define the "key" line segments and eventually form the new PaLiS representation of road graphs. We assume that the patch size p is divisible by H and W without loss of generality.

The Main Representation

By generating a set of N non-overlapping $p \times p$ patches $\{\mathcal{P}_i\}$ where $N = \frac{H}{p} \times \frac{W}{p}$, we define the patched line segment for each local patch $\mathcal{P}_i$. As shown in Fig. 3(b), there are three cases for each patch $\mathcal{P}_i$ depending on the number of roads passing through the patch, denoted by $\mathcal{N}(\mathcal{P}_i) \in \mathbb{N}$. If $\mathcal{N}(\mathcal{P}_i) = 0$, we term it as the background patch (*i.e.*, the gray patches in Fig. 3). If $\mathcal{N}(\mathcal{P}_i) = 1$, we uniquely define its patched line segment, denoted by

$$\text{PaLiS}(\mathcal{P}_i) = (x_i^u, y_i^u, x_i^v, y_i^v) \in \mathbb{R}^4 \text{ if } \mathcal{N}(\mathcal{P}_i) = 1. \quad (1)$$

For those patches that satisfy $\mathcal{N}(\mathcal{P}_i) > 1$, we cannot uniquely define their line segments, but we found such patches are playing a key role to construct the expected road graphs. As shown in Fig. 4, we further study the properties of the patches that have $\mathcal{N}(\mathcal{P}_i) \geq 1$. In Fig. 4(a), the foreground patches clearly define a (local) straight road without ambiguity. But for the patches that have $\mathcal{N}(\mathcal{P}_i) > 1$, there are two types as shown in Fig. 4(b) and 4(c), depending on if there is an annotated "keypoint" to connect the multiple road paths in one keypoint. If there is such keypoint annotation, we call such a type as the X-type. Otherwise, the multiple road paths passing through the patch $\mathcal{P}_i$ will have different elevations like the overpasses, and we called them as the T-type patches.

In summary, the proposed PaLiS representation firstly samples N non-overlapping local patches and identifies the foreground patches by three different types, the I-type, X-type and T-type, and defines the local line segments for the I-type patches in the form of $(x_i^u, y_i^u, x_i^v, y_i^v)$ to retain the geometric information of road paths. In the next section, we will show how to learn our proposed PaLiS representation for the task of vector road mapping.

Road Graph Reconstruction from PaLiS

Thanks to our geometric PaLiS representation, the road graphs can be reasonably reconstructed without leveraging

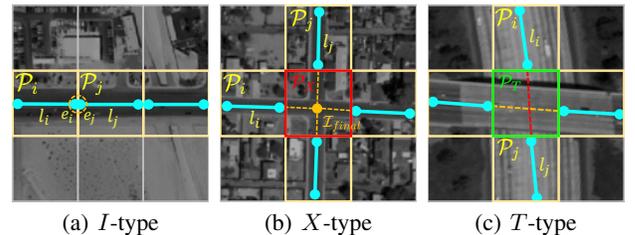(a) I-type   (b) X-type   (c) T-type

Figure 4: Illustration of different types of foreground patches. Patched line segments are denoted in cyan markers and connectivities are denoted in dashed markers.
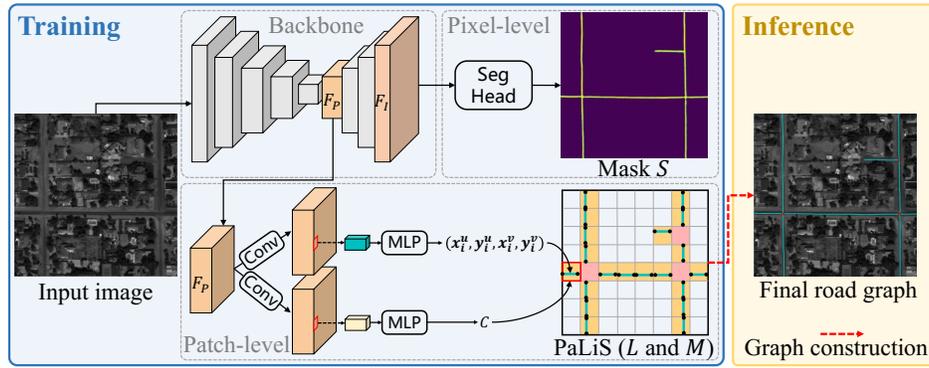
Figure 5: Proposed Method Pipeline: The process begins with an input image. (1) An Encoder-Decoder network extracts pixel-level ($\mathbf{F_I}$) and patch-level ($\mathbf{F_P}$) feature maps. (2) The Patch-level branch uses $\mathbf{F_P}$ to predict line segments (coordinates $(x_i^u, y_i^u, x_i^v, y_i^v)$) and patch types ($C$). (3) The Pixel-level branch generates a binary mask of road centerlines from $\mathbf{F_I}$. (4) Finally, a road graph is reconstructed using the PaLiS representation. Note: Enlarged patches are shown for clarity.

another subnetwork for the learning of graph connectivity. Here, we hypothesize that the PaLiS representation can be reliably learned and defer the learning details in Sec. 4. We developed a geometrically-meaningful scheme to reconstruct the road graphs from our PaLiS representation (*see our supp. material for the pseudo code*) by considering the properties of I-type, X-type and T-type foreground patches in the following three cases:

- As shown in Fig. 4(a), we first consider the most common case for the I-type patches. For two adjacent I-type patches $\mathcal{P}_i$ and $\mathcal{P}_j$, their line segments $\mathbf{l}_i = \mathrm{PaLiS}(\mathcal{P}_i)$ and $\mathbf{l}_j = \mathrm{PaLiS}(\mathcal{P}_j)$ are connected with the observation that line segments of adjacent I-type patches share a common endpoint. We formulate the rule based on the shape distance $\mathrm{d_s}(A, B)$, which represents the shortest perpendicular distance between shapes A and B. Two line segments are connected if the average of $\mathrm{d_s}(\mathbf{l}_j, \mathbf{e}_i)$ and $\mathrm{d_s}(\mathbf{l}_i, \mathbf{e}_j)$ is less than a given distance threshold τ_d, where $\mathbf{e}_i$ is the endpoint of $\mathbf{l}_i$ close to the line segment $\mathbf{l}_j$ and $\mathbf{e}_j$ is the endpoint of $\mathbf{l}_j$ close to the line segment $\mathbf{l}_i$.

- While encountering the X-type patch $\mathcal{P}_X$ (*e.g.*, cross roads), line segments surrounding the patch $\mathcal{P}_X$ are extended to an intersection as shown in Fig. 4(b). To achieve this, candidate intersections are calculated by pairing up lines segments around the patch $\mathcal{P}_X$. The intersection $\mathcal{I}_{i,j} \in \mathcal{R}^2$ of the line segment pair $(\mathbf{l}_i, \mathbf{l}_j)$ is valid if the two line segments intersect within the patch $\mathcal{P}_X$. And the final intersection $\mathcal{I}_{final}$ is obtained by averaging the position of all candidate intersections and is connected to the surrounding line segments.

- Regarding the T-type patch $\mathcal{P}_T$ (*e.g.*, overpasses), the layouts with different height are made by the directional and spatial and extension of roads as shown in Fig. 4(c). We pair up lines segments around the patch $\mathcal{P}_T$ and the connection of a line segments pair $(\mathbf{l}_i, \mathbf{l}_j)$ is valid if the shape distance $\mathrm{d_s}(\mathbf{l}_i, \mathbf{l}_j)$ and the angle difference $\mathrm{d}_{angle}(\mathbf{l}_i, \mathbf{l}_j)$ are less than the distance threshold τ_d and the angle threshold τ_a respectively.

4 Learning PaLiS Representations

In this section, we show how to reliably learn the proposed PaLiS representation for vector road mapping in an off-the-shelf ConvNet. We use an encoder-decoder network, DLinkNet (Zhou, Zhang, and Wu 2018), with the lightweight ResNet-34 (He et al. 2016) as the backbone encoder to extract feature maps for the learning of PaLiS. Fig. 5 shows the overall pipeline of our approach. For the learning of PaLiS representation, two headnets are respectively leveraged, to classify the patches according to their PaLiS classes, and regress the two endpoints for each I-type patch. Apart from the main branches, an auxiliary segmentation head is leveraged to learn the rasterized masks from the final feature maps of the decoder network.

Identifying Patch Classes/Types

Our PaLiS representation categorizes the foreground patches into three different types (I-type, X-type, and T-type) for a better understanding of intricate road graph structures. To achieve this, we use a patch classification head, which consists of four convolution layers all with 3×3 kernels and an MLP layer, to predict the class of each patch. The patch classification head takes patch-level feature maps $\mathbf{F_P}$ as input and produces the patch map $\mathbf{M} \in \mathbb{R}^{C_P \times \frac{H}{P} \times \frac{W}{P}}$, where C_P is the number of patch classes (*i.e.*, $C_P = 4$ by considering the background patches). During training, we compute the classification loss by comparing the predicted patch map $\mathbf{M}$ with the corresponding ground truth $\mathbf{M}^*$ which can be easily obtained from the original annotations of the dataset. Cross-entropy loss is employed for $\mathbf{M}$:

$$\mathcal{L}_{\mathcal{M}} = \mathrm{CE}(\mathbf{M}, \mathbf{M}^*). \qquad (2)$$

Line Segments Learning for I-type Patches

With the patch classification head, we focus on the I-type patches to learn the patched line segments. It should be noted that although the line segment $\mathbf{l}_i$ for the patch $\mathcal{P}_i$ is in the closed-form for the two endpoints, directly regressing their endpoint coordinates is suboptimal since the data augmentation techniques (like cropping) used in the training phase

Model	Type	City-Scale				SpaceNet			
			TOPO		APLS ↑		TOPO		APLS ↑
		P ↑	R ↑	F1 ↑		P ↑	R ↑	F1 ↑	
DLinkNet (Zhou, Zhang, and Wu 2018)	Mask	78.63	48.07	57.42	54.08	88.42	60.06	68.80	56.93
Orientation (Batra et al. 2019)		75.83	68.90	72.20	55.34	81.56	71.38	76.13	58.82
Seg-DLA (He et al. 2020)		75.59	72.26	73.89	57.22	78.99	69.80	74.11	56.36
RoadTracer (Bastani et al. 2018)	Point	78.00	57.44	66.16	57.29	78.61	62.45	69.90	56.03
Sat2Graph (He et al. 2020)		80.70	72.28	76.26	63.14	85.93	76.55	80.97	64.43
TD-Road (He, Garg, and Chowdhury 2022)		81.94	71.63	76.27	65.74	84.81	77.80	81.15	65.15
PaRK-Detect (Xie et al. 2023)		82.17	68.23	74.29	67.66	91.34	68.07	78.01	62.97
RNGDet (Xu et al. 2022)		85.97	69.78	76.87	65.75	90.91	73.25	81.13	65.61
RNGDet++ (Xu et al. 2023b)		85.65	72.58	78.44	67.76	**91.34**	75.24	82.51	67.73
Ours	PaLiS	**86.36**	**73.16**	**79.08**	**68.12**	90.05	**78.19**	**83.70**	**69.68**

Table 1: Quantitative results on City-scale Dataset and SpaceNet dataset. Best results are highlighted in bold.
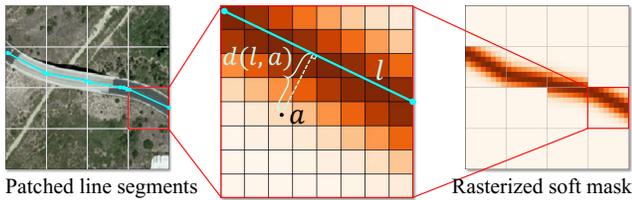

Figure 6: Illustration of the rasterization. Darker pixels contribute more to the line segment.

will incur inefficient computation in terms of cropping the vector road annotations. To avoid this issue, we propose to use the differentiable rasterization techniques to learn the line segment $\mathbf{l}_i$ of the patch $\mathcal{P}_i$ from the mask supervision, similar to (Lazarow, Xu, and Tu 2022; Zorzi et al. 2022). It is interesting to see that, although we use the rasterized road mask supervision instead of the vector annotations, such a design is prevailing than the vector annotations. Please move to our ablation studies in Sec. 5 for a detailed comparison.

By taking the feature map $\mathbf{F}_P$, we set a regression head with four 3×3 convolution layers and an MLP layer, to predict line segments $\mathbf{L} \in \mathbb{R}^{4 \times \frac{H}{p} \times \frac{W}{p}}$ where 4 is the number of coordinates of line segments. These patched line segments $\mathbf{L}$ are then converted into a soft mask $\mathbf{S_{soft}} \in \mathbb{R}^{\mathbf{H} \times \mathbf{W}}$ with the proposed rasterizer. As shown in Fig. 6, the proposed rasterizer produces a $p \times p$ patch $\mathbf{C}_i \in \mathbb{R}^{p \times p}$, where the scalar value at the pixel $\mathbf{a} = (x, y)$ in the local coordinate of the patch is computed by

$$\mathbf{C}_i(\mathbf{a}) = e^{\frac{-\mathrm{d}^2(\mathbf{l}_i, \mathbf{a}) \times t}{\tau_{\mathrm{inv}}}}, \tag{3}$$

where $\mathrm{d}(\mathbf{l}_i, \mathbf{a})$ is the projection distance from the pixel $\mathbf{a}$ to the line segment $\mathbf{l}_i$. t and τ_{inv} are the projection factor and sharpness factor respectively. We empirically set $t = 10$ if the pixel $\mathbf{a}$ is projected outside of the line segment otherwise t is set to 1. The values of projection factor t and the sharpness factor τ_{inv} are chosen to accurately reflect the position of the line segment in the patch.

The rasterized soft mask $\mathbf{S_{soft}} \in \mathbb{R}^{\mathbf{H} \times \mathbf{W}}$ is obtained from the contributions of all pixels. During training, we efficiently

compute the loss by comparing the soft mask $\mathbf{S_{soft}}$ with the existing ground truth mask $\mathbf{S}^*$ of road centerlines. Similar to BoundaryFormer (Lazarow, Xu, and Tu 2022), we employ the DICE (Milletari, Navab, and Ahmadi 2016) loss to measure the difference:

$$\mathcal{L}_{\mathcal{L}} = \mathrm{DICE}(\mathbf{S_{soft}}, \mathbf{S}^*). \tag{4}$$

The rasterizer and backwards pass are fully implemented in CUDA, ensuring efficiency in the training process.

Auxiliary Pixel-level Learning

In addition to the PaLiS representation, we incorporate the learning of an auxiliary binary mask for road centerlines to extract road information. We use a segmentation head, which consists of one 3×3 convolution layer and one 1×1 convolution layer followed by a sigmoid function, to predict the binary mask $\mathbf{S} \in \mathbb{R}^{H \times W}$ of road centerlines from the pixel-level feature maps $\mathbf{F_I}$. We compute the loss of the predicted binary mask $\mathbf{S}$ with the ground truth mask $\mathbf{S}^*$ of road centerlines by cross-entropy loss:

$$\mathcal{L}_{\mathcal{S}} = \mathrm{CE}(\mathbf{S}, \mathbf{S}^*) \tag{5}$$

The total loss of the PaLiS learning can be summarized as

$$\mathcal{L}_{total} = \mathcal{L}_{\mathcal{S}} + \mathcal{L}_{\mathcal{M}} + \mathcal{L}_{\mathcal{L}}. \tag{6}$$

5 Experiments

In this section, we run experiments for our proposed PaLiS-based approach on public benchmarks and provide a comprehensive analysis of our design choices. The implementation details are in our supplementary material.

Datasets and Evaluation Metrics

Datasets. We conduct experiments on two widely used datasets: City-Scale dataset (He et al. 2020) and SpaceNet dataset (Van Etten, Lindenbaum, and Bacastow 2018). City-Scale dataset (He et al. 2020) covers $720 \ km^2$ area of 20 cities in the United States. It consists of 180 tiles, which we divide into 144, 9, and 27 tiles for training, validation, and testing respectively, following previous methods (He et al.

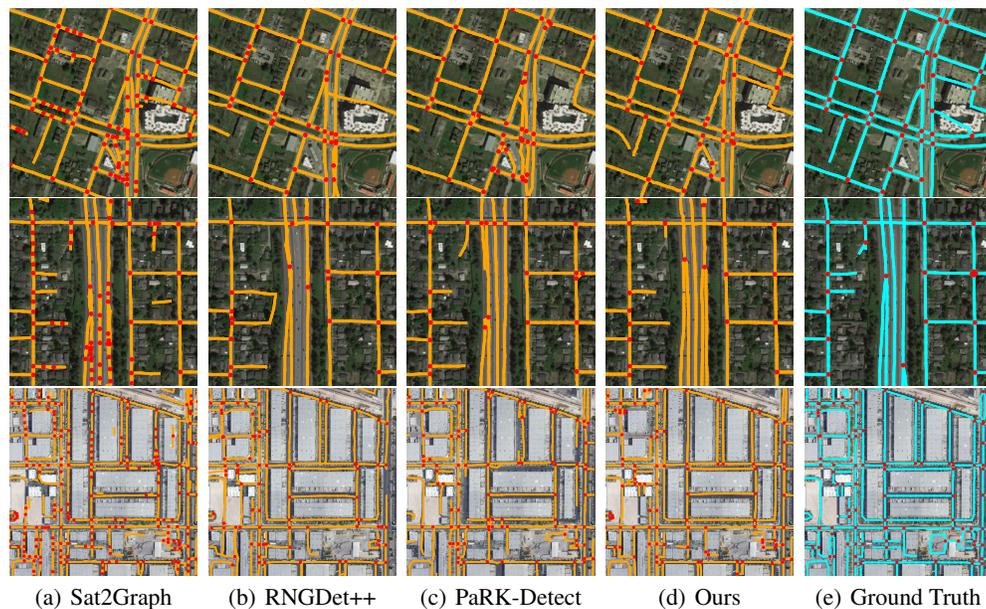| (a) Sat2Graph | (b) RNGDet++ | (c) PaRK-Detect | (d) Ours | (e) Ground Truth |

Figure 7: Example of qualitative road network extraction results on City-scale dataset. Predicted road segments are marked in orange lines and intersections are marked in red dots. Our approach leads to reasonable and accurate connected road graphs.

2020; He, Garg, and Chowdhury 2022; Xu et al. 2023b). Each tile of the dataset has the resolution of 2048×2048 pixels, representing 1 meter in the real world. SpaceNet dataset (Van Etten, Lindenbaum, and Bacastow 2018) comprises 2549 satellite images, each with the resolution of 400×400 pixels. We use 2040, 127, and 382 images for training, validation, and testing respectively, following the partition used in Sat2Graph (He et al. 2020).

Evaluation metrics. Two quantitative metrics are utilized in the experiments: APLS (Van Etten, Lindenbaum, and Bacastow 2018) and TOPO (Biagioni and Eriksson 2012). APLS assesses the overall graph quality by comparing the similarity of shortest paths between two locations on the predicted and ground truth graphs. On the other hand, the TOPO metric (precision, recall, and F1-score) provides a stricter evaluation of detailed topology correctness by measuring the similarity of sub-graphs sampled from a seed location on the ground truth and predicted graphs. Higher scores indicate better performance for both APLS and TOPO metrics.

Main Comparisons

Quantitative and Qualitative Evaluation. We compare our approach to state-of-the-art segmentation- and keypoint-based methods on the City-Scale and SpaceNet datasets. Table 1 presents the quantitative results. Segmentation-based methods exhibit substantially inferior performance on both TOPO and APLS metrics, because of their heuristic post-processing schema. In contrast, graph-based methods output and refine the graph of road networks directly, gaining better performance on the two metrics. Our method achieves the highest TOPO and APLS scores on the City-Scale dataset, demonstrating superior performance in capturing road net-

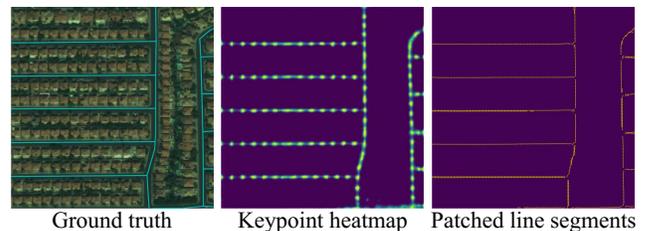| Ground truth | Keypoint heatmap | Patched line segments |

Figure 8: Comparison on early stage (10 epochs) of training.

work structures with our unified PaLiS representations. Additionally, our approach outperforms all other methods in terms of recall, F1-score, and APLS on SpaceNet dataset, further validating its effectiveness. These consistently superior evaluation results across metrics indicate that our approach generates more precise and complete road graphs both locally and globally. The same conclusions can be drawn from the qualitative comparisons in Fig. 7.

Keypoint-based Graph Representation *v.s.* PaLiS. Comparisons between keypoints and PaLiS representation during training and testing involved further analysis. Fig. 8 first visualized the predicted keypoints heatmap and line segments on the early training epoch. Apparently, the learned keypoints heatmap was ambiguous in the early stage of training, whereas the line segments were accurately predicted. Subsequently, we studied the model's sensitivity to thresholds of keypoints (or line segments) prediction by varying the thresholds with the 0.1 step as shown in Fig. 9. Notably, our model demonstrated greater stability compared to keypoint-based methods, indicating the robustness of our PaLiS representation during testing.
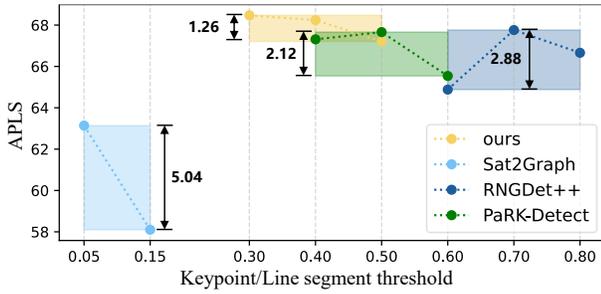
Figure 9: Parameter sensitivity on City-Scale dataset.

Supervision	TOPO			APLS ↑
	P ↑	R ↑	F1 ↑	
unsorted vector	91.79	60.34	72.82	57.34
sorted vector	91.75	66.81	77.31	65.28
raster mask	90.05	78.19	83.70	69.68

Table 2: Comparison results on SpaceNet dataset in association with different supervisions for patched line segments.

Training efficiency. The training efficiency is also compared as shown in Fig. 2. The approach relying on our unified PaLiS representation achieves superior performance with considerably fewer training iterations while methods relying on keypoints (He et al. 2020; Xu et al. 2023b; Xie et al. 2023) require much more iterations to converge.

Ablation Studies

Mask-supervised line segment learning. To evaluate the efficacy of the proposed soft rasterizer, we conducted additional experiments using three different types of supervision for line segments learning: unsorted vector labels, sorted vector labels, and mask labels. The unsorted and sorted vector labels are denoted by $(\hat{x}_i^u, \hat{y}_i^u, \hat{x}_i^v, \hat{y}_i^v) \in \mathbb{R}^4$, where the only difference is the direction. Directions of unsorted vector labels are randomly inherent from origin annotations, while sorted vector labels have consistent directions ($(\hat{x}_i^u, \hat{y}_i^u)$ is always the endpoint on the left). We use L1 loss to compute the difference between the predictions and ground truth vector labels. The results shown in Table 2 indicate that line segments are learned more precisely with the proposed rasterizer, leading to enhanced connectivity in the graph construction. Furthermore, our approach leverages the existing mask labels to guide the training process of patched line segments, without requiring the generation of vector labels.

Graph construction strategy. Road graphs can be reconstructed by PaLiS representation (geometric connectivity) without the learned relationships of patches (relationship connectivity) used in PaRK-Detect (Xie et al. 2023). To compare the two different construction strategies, we learned additional relationships of patches following PaRK-Detect (Xie et al. 2023). The results presented in Table 3 show that our approach outperforms the relationship connectivity on the two metrics, and provides more accurate and

Graph construction	TOPO			APLS ↑
	P ↑	R ↑	F1 ↑	
Learned relationship	88.01	79.28	83.42	68.47
Ours	90.05	78.19	83.70	69.68

Table 3: Results on SpaceNet of varied connectivity strategy.

Patch Size	TOPO			APLS ↑
	P ↑	R ↑	F1 ↑	
4×4	91.88	74.23	82.12	67.25
8×8	90.05	78.19	83.70	69.68
16×16	82.61	77.64	80.05	67.58

Table 4: Results on Spacenet of varied patch size.

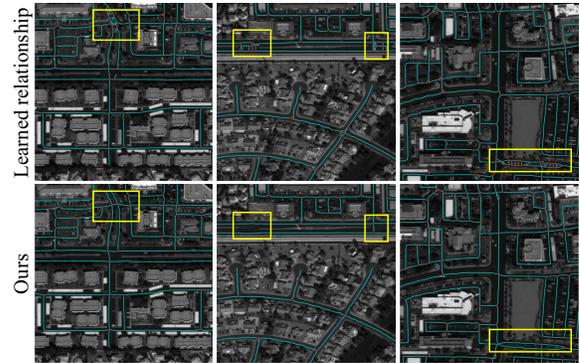reasoned connectivity as shown in Fig. 10.

Figure 10: Comparison of graph construction strategies.

Patch size. The patch size serves as a crucial hyperparameter in our approach. We conducted experiments to assess the impact of patch size, and the results are shown in Table 4. We observe that both smaller and bigger patch sizes cause the inferior performance. This is due to the PaLiS representation with small patch size yields results that are close to mask representation, suffering from the disconnected issue. Whereas PaLiS representation with big patch size struggles to provide precise shape of road graphs. Considering accuracy and efficiency, we set the patch size to 8.

6   Conclusions

This paper introduces a learning-based approach for vector road mapping using the innovative PaLiS (Patched Line Segment) representation. By leveraging local patches, our approach effectively represents road graphs. Through convolutional neural networks, we achieve state-of-the-art performance on public datasets, with efficient training in just 6 GPU hours. Additionally, the ability of PaLiS representation to learn line segment endpoint coordinates from rasterized road maps suggests a promising direction for large-scale vector road mapping without costly manual annotations in the near future.

Acknowledgments

References

Bastani, F.; He, S.; Abbar, S.; Alizadeh, M.; Balakrishnan, H.; Chawla, S.; Madden, S.; and DeWitt, D. 2018. Road-tracer: Automatic extraction of road networks from aerial images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 4720–4728.

Batra, A.; Singh, S.; Pang, G.; Basu, S.; Jawahar, C.; and Paluri, M. 2019. Improved road connectivity by joint learning of orientation and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 10385–10393.

Biagioni, J.; and Eriksson, J. 2012. Inferring road maps from global positioning system traces: Survey and comparative evaluation. *Transportation Research Record*, 2291(1): 61–71.

Büchner, M.; Zürn, J.; Todoran, I.-G.; Valada, A.; and Burgard, W. 2023. Learning and Aggregating Lane Graphs for Urban Automated Driving. arXiv:2302.06175.

Cai, Z.; Wang, T.; Mi, Q.; Su, X.; Guo, L.; and Ding, Z. 2023. Dynamic Weighted Road Network Based Multi-Vehicles Navigation and Evacuation. *ISPRS International Journal of Geo-Information*, 12(3): 127.

Cheng, M.; Zhao, K.; Guo, X.; Xu, Y.; and Guo, J. 2021. Joint topology-preserving and feature-refinement network for curvilinear structure segmentation. In *Int. Conf. Comput. Vis.*, 7147–7156.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 770–778.

He, S.; and Balakrishnan, H. 2022. Lane-Level Street Map Extraction from Aerial Imagery. In *IEEE/CVF Wint. Conf. on Appl. of Comp. Vis.*, 1496–1505.

He, S.; Bastani, F.; Jagwani, S.; Alizadeh, M.; Balakrishnan, H.; Chawla, S.; Elshrif, M. M.; Madden, S.; and Sadeghi, M. A. 2020. Sat2graph: Road graph extraction through graph-tensor encoding. In *Eur. Conf. Comput. Vis.*, 51–67. Springer.

He, Y.; Garg, R.; and Chowdhury, A. R. 2022. TD-Road: Top-down road network extraction with holistic graph construction. In *Eur. Conf. Comput. Vis.*, 562–577. Springer.

Lazarow, J.; Xu, W.; and Tu, Z. 2022. Instance segmentation with mask-supervised polygonal boundary transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 4382–4391.

Li, T.-M.; Lukáč, M.; Gharbi, M.; and Ragan-Kelley, J. 2020. Differentiable Vector Graphics Rasterization for Editing and Learning. *ACM Trans. Graph.*, 39(6).

Máttyus, G.; Luo, W.; and Urtasun, R. 2017. Deeproadmapper: Extracting road topology from aerial images. In *Int. Conf. Comput. Vis.*, 3438–3446.

Mei, J.; Li, R.-J.; Gao, W.; and Cheng, M.-M. 2021. CoANet: Connectivity attention network for road extraction from satellite imagery. *IEEE Trans. Image Process.*, 30: 8540–8552.

Milletari, F.; Navab, N.; and Ahmadi, S.-A. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *IEEE Int. Conf. 3D Vis.*, 565–571. Ieee.

Shi, G.; Shan, J.; Ding, L.; Ye, P.; Li, Y.; and Jiang, N. 2019. Urban road network expansion and its driving variables: a case study of Nanjing City. *International Journal of Environmental Research and Public Health*, 16(13): 2318.

Shit, S.; Koner, R.; Wittmann, B.; Paetzold, J.; Ezhov, I.; Li, H.; Pan, J.; Sharifzadeh, S.; Kaissis, G.; Tresp, V.; et al. 2022. Relationformer: A unified framework for image-to-graph generation. In *Eur. Conf. Comput. Vis.*, 422–439. Springer.

Van Etten, A.; Lindenbaum, D.; and Bacastow, T. M. 2018. Spacenet: A remote sensing dataset and challenge series. *arXiv preprint arXiv:1807.01232*.

Wang, B.; Liu, Q.; Hu, Z.; Wang, W.; and Wang, Y. 2023. TERNformer: Topology-enhanced Road Network Extraction by Exploring Local Connectivity. *IEEE Trans. Geosci. Remote. Sens.*

Xie, G.; Sun, X.; Tong, X.; and Nowrouzezahrai, D. 2014. Hierarchical diffusion curves for accurate automatic image vectorization. *ACM Trans. Graph.*, 33(6): 230:1–230:11.

Xie, S.; Zheng, W.; Xian, Z.; Yang, J.; Zhang, C.; and Wu, M. 2023. PaRK-Detect: Towards Efficient Multi-Task Satellite Imagery Road Extraction via Patch-Wise Keypoints Detection. In *Brit. Mach. Vis. Conf.*

Xu, B.; Xu, J.; Xue, N.; and Xia, G.-S. 2023a. HiSup: Accurate polygonal mapping of buildings in satellite imagery with hierarchical supervision. *ISPRS Journal of Photogrammetry and Remote Sensing*, 198: 284–296.

Xu, Z.; Liu, Y.; Gan, L.; Sun, Y.; Wu, X.; Liu, M.; and Wang, L. 2022. Rngdet: Road network graph detection by transformer in aerial images. *IEEE Trans. Geosci. Remote. Sens.*, 60: 1–12.

Xu, Z.; Liu, Y.; Sun, Y.; Liu, M.; and Wang, L. 2023b. RNGDet++: Road Network Graph Detection by Transformer With Instance Segmentation and Multi-Scale Features Enhancement. *IEEE Robotics Autom. Lett.*

Xu, Z.; Sun, Y.; and Liu, M. 2021. icurb: Imitation learning-based detection of road curbs using aerial images for autonomous driving. *IEEE Robotics and Automation Letters*, 6(2): 1097–1104.

Xue, N.; Bai, S.; Wang, F.; Xia, G.; Wu, T.; Zhang, L.; and Torr, P. H. S. 2021. Learning Regional Attraction for Line Segment Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(6): 1998–2013.

Xue, N.; Bai, S.; Wang, F.; Xia, G.-S.; Wu, T.; and Zhang, L. 2019. Learning attraction field representation for robust line segment detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 1595–1603.

Xue, N.; Wu, T.; Bai, S.; Wang, F.; Xia, G.; Zhang, L.; and Torr, P. H. S. 2022. Holistically-Attracted Wireframe Parsing: From Supervised to Self-Supervised Learning. *arXiv*, abs/2210.12971.

Xue, N.; Wu, T.; Bai, S.; Wang, F.; Xia, G.-S.; Zhang, L.; and Torr, P. H. 2020. Holistically-attracted wireframe parsing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2788–2797.

Yang, B.; Zhang, M.; Zhang, Z.; Zhang, Z.; and Hu, X. 2023. TopDiG: Class-Agnostic Topological Directional Graph Extraction From Remote Sensing Images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 1265–1274.

Zhang, K.; Zhao, D.; Feng, L.; and Cao, L. 2021. Cycling trajectory-based navigation independent of road network data support. *ISPRS International Journal of Geo-Information*, 10(6): 398.

Zhou, L.; Zhang, C.; and Wu, M. 2018. D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 182–186.

Zorzi, S.; Bazrafkan, S.; Habenschuss, S.; and Fraundorfer, F. 2022. Polyworld: Polygonal building extraction with graph neural networks in satellite images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 1848–1857.