TopoSense: agent driven topological graph extraction from remote sensing image

Mi Zhang[1, 3, †], Bingnan Yang[1], Jianya Gong[1,2], Xiangyun Hu[1, 3]

[1] School of Remote Sensing and Information Engineering, Wuhan University,
No.129, Luoyu Road, Wuhan 430079, China. † mizhang@whu.edu.cn
[2] State Key Laboratory of Information Enginnering in Surveying, Mapping and Remote Sensing, Wuhan University,
No.129, Luoyu Road, Wuhan 430079, China.
[3] Hubei Luojia Laboratory, No.129, Luoyu Road, Wuhan 430079, China.

Keywords: Topological graph extraction, Remote sensing image interpretation, Agent-driven representation, Reinforcement learning, Topological connectivity enhancement, Collaborative optimization.

Abstract

Automatic topological graph extraction is critical for intelligent remote sensing image interpretation and cartographic representation. However, existing approaches neither adopt segmentation-based post-processing nor directly predict the graph, thereby suffering from limited scalability and poor adaptability to complex spatial structures. To address these issues, we introduce TopoSense, an innovative framework for extracting topological graphs from remote sensing images through an agent-driven approach. By employing a novel combination of reinforcement learning and neural network architectures, TopoSense autonomously navigates through pixel-level data, efficiently constructing topological representations. It not only enhances the accuracy of spatial feature detection, but also significantly reduces processing time. Experiments on the TOP-BOUNDARY and REALSCENE demonstrate its superiority in capturing intricate spatial relationships compared to traditional methods.

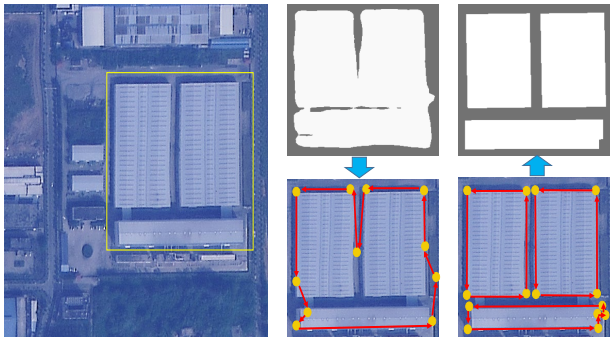
1. INTRODUCTION

Topological graph extraction (TGE) is essential in intelligent remote sensing image interpretation, aiming to identify the semantic information of geographical objects and precisely reconstruct their topological connections. The advantages of employing vector representations for these topological graphs include minimal redundancy, ease of topological analysis, and enhanced accuracy in geographic location queries. TGE has been instrumental in various applications, including automated mapping (Liu et al., 2023b), emergency response for disaster mitigation (Zorzi et al., 2020), and the creation of high-definition maps for autonomous vehicles (Chen et al., 2023). Consequently, a variety of TGE methods have been developed, with segmentation-based (Wei et al., 2020) and graph-based (Belli and Kipf, 2019) approaches being the most prevalent.

Segmentation-based TGE begin with the creation of a semantic segmentation map, subsequently utilizing post-processing techniques, such as skeletonization and binarization, to refine and extract the topological graph. It typically employs CNN or Transformer (Vaswani et al., 2017) as the backbone and refine it by conducting Douglas–Peucker (Douglas and Peucker, 1973) simplification. For example, GGT leverages the Transformer model to iteratively predict nodes and their connections based on road segmentation results, ultimately establishing a vector topological structure. Similarly, tools like PolyMapper and ASIP employ Recurrent Neural Networks and semantic polygon decomposition methods, respectively, to enhance segmentation outcomes. Hatamizadeh et al. (Hatamizadeh et al., 2020) and Girard et al. (Girard et al., 2021) have explored the use of active contour models with orientation field constraints to improve the representation of building topologies. However, approaches like those of Wei et al. (Wei et al., 2019), which focus on simplifying and regularizing building contours, have been observed to adversely affect boundary precision, as measured by the Intersection over Union (IoU) metric. A common challenge faced by these techniques is preserving the topological integrity, especially in complex urban landscapes with numerous intersecting or overlapping elements.

Another TGE paradigm advocates the usage of graph-based representation. It leverages keypoints to reconstruct the topological structures and represent it as a directed acyclic graph (DAG). This approach, exemplified by RoadTracer (Bastani et al., 2018), employs iterative methods to construct DAGs of centerlines by predicting keypoints and their decision actions. Likewise, the incremental learning method introduced by (Lian and Huang, 2020), along with RNGDet series (Xu et al., 2022, Xu et al., 2023) that employs Transformer models, concentrate on the adjacency of keypoints to define the vector topological structure. Graph-based TGE also simultaneously predicts keypoints and their connectivity relationships. APGA (Zhu et al., 2021) fall into this paradigm, which learns both the positions of keypoints and their topological connectivity relations, such as angle trends. Further innovations include the iCurb series (Xu et al., 2021a), which uses keypoints' adjacency matrices for road prediction, and the PolyWorld model, which employs optimal transport methods to generate DAGs. Liu et al. (Liu et al., 2023a) built the PolyFormer model, treating objects in natural images as sequences of coordinate points and obtaining target boundaries through regression of coordinate sequences. Although these methods enhance the accuracy of topological structure predictions through iterative or incremental approaches, the occurrence of multiple starting points often leads to the generation of redundant pathways, thereby diminishing the efficiency of the iterative process. Successfully amalgamating global and local topological insights remains a significant challenge.

As a specialized type of graph-based representation, contour-based methods focus on instance-level adoption of initial contours. Initially, contours are derived from segmentation outcome processing or deformations of object detection boxes. Sub-



(a) Source Image (b) Traditional method (c) TopoSense

Figure 1. Illustration of traditional method and TopSense.

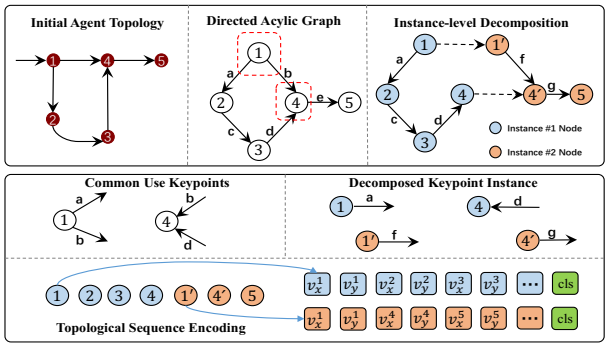


Figure 2. Keypoint driven topological graph represenation.

sequently, the sampling points on these initial contours undergo optimization through techniques such as circular convolution (Yan et al., 2021) and graph neural networks, aiming to refine the target boundary graph. The PolarMask network (Xie et al., 2020) extracts a rough graph instance via instance center classification and dense distance regression within a polar coordinate system. Curve-GCN (Ling et al., 2019) implements GCN and conceptualizes each instance as a circle of control points. (Wei et al., 2023) utilizes the bounding boxes of instances as initial contours for areal features, applying Curve-GCN to aerial imagery for building extraction. Another study (Peng et al., 2020) shapes initial contours as octagons by deforming object detection boxes, using circular convolution on uniformly sampled contour points to achieve a refined instance-level topological graph. Additionally, the SharpContour network (Zhu et al., 2022) enhances graph precision by predicting offsets of initial contour points for planar features. These methods simplify the process by eliminating complex post-processing steps and facilitating straightforward establishment of topological relationships. However, their suitability is limited to instance-level feature extraction and they struggle with polyline structures featuring complex topological relationships.

Given the outlined limitations, we introduce a novel TGE framework named *TopoSense, which leverages keypoint-based agent to derive topological graphs*. Figrue 1 illustrates the boundary effects managed by keypoint-based strategies. Unlike traditional segmentation-based and graph-based methods, TopoSense effectively addresses the common negative boundary effects seen in segmentation-based methods, reduces path redundancy in iterative or incremental approaches, and surmounts the challenges associated with extending instance-level TGE to more diverse geospatial feature types. Firstly, to alleviate the negative boundary effects, TopoSense treats keypoints located along the primary boundary or centerline as an agent, optimizing the linkage of points through reinforced historical exploration. This strategy ensures more accurate boundary delineation. Secondly, to enhance the efficiency of our iterative graph-building process, we have integrated a topology memory-replay module that significantly reduces redundant path computations, thereby streamlining the graph construction process. Finally, by decoupling the prediction of feature types into type-independent tasks involving the identification of keypoints and their connections, TopoSense utilizes the keypoint agent's capacity to model both local and global topological relationships. This adaptability makes TopoSense broadly applicable in general TGE, enhancing its utility across a variety of geospatial contexts.

On a glance, we deliver the following contributions:

- We revisit and compare different TGE paradigms and present the an agent driven *TopoSense* framework.
- The universal negative boundary alleviation and topology memory-replay modules are proposed, with a Universal-Topology Hub (UT-Hub) for depicting keypoint connection and a Topology Memory MiXer (TMX) for replaying historical cues.
- We investigate the TopoSense generalization capacity by applying it to geospatial polygon and polyline tyeps and notice that TopoSense achieves state-of-the-art performances on the different types of datasets.

2. OUR APPROACH

In this section, we will elaborate the overall TopoSense architecture in Sec. 2.1, the Universal-Topology Hub in Sec. 2.2 and Topology Memory Mixer for reducing path redundancy in Sec. 2.3.

2.1 TopoSense Architecture

We first introduce the representation of agent's topology graph and then show the main components in TopoSense.

Agent Topology Representation. Figure 2 depicts the Directed Acyclic Graph (DAG) representation for the keypoint-driven agent. Considering that polyline and polygon targets, such as roads and buildings, are composed of keypoints, and that predictions for centerlines or boundaries essentially involve predicting a DAG, we treat the predictions for these types of targets as the construction of a DAG. Firstly, keypoints from the centerline or boundary serve as the initial agents, with their topology constructed from the connections between keypoints, *i.e.*, a global DAG. Secondly, we decompose this global DAG into instance-level segments using commonly shared keypoints, for example, nodes #1 and #4 in Figure 2.. Given that keypoints align with these instance-level segments, the formulated DAG, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consists of a vertex set $\mathcal{V}$, representing instance-level nodes, and an edge set $\mathcal{E}$, representing all instance-level segments. Each vertex $v = (v_x, v_y) \in \mathcal{V}$ contains two properties: (i) location, *i.e.*, $(v_x \in \mathbb{R}, v_y \in \mathbb{R})$, (ii) category, *i.e.*, cls. While each edge $e = (e_s, e_t) \in \mathcal{E}$ is composed of the source keypoint e_s and target keypoint e_t. The decomposed DAG instance is finaly utilized for topological sequence encoding in the UT-Hub.

Architecture components. In Figure 3, our TopoSense system comprises three principal components: the image feature extraction backbone, UT-Hub, and the Topology Memory Mixer

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-3-2024-445-2024 | © Author(s) 2024. CC BY 4.0 License.

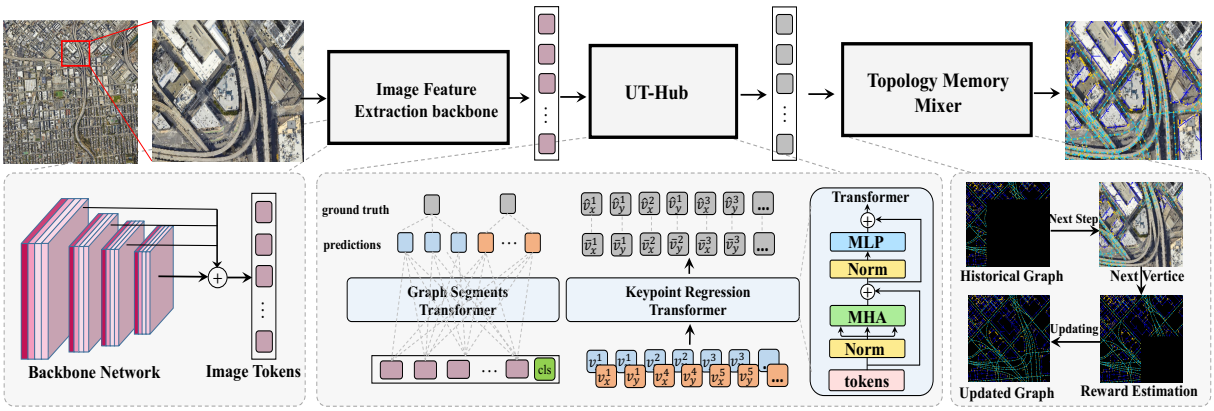


Figure 3. Overall architecture of TopSense. The system comprises three main modules: the image feature extraction backbone, UT-Hub, and the Topology Memory MiXer (TMX). The first two predict keypoints and graph segments, respectively, while TMX focuses on forecasting keypoint connections by dynamically updating the historical graph.

(TMX). The image feature extractor processes a cropped region denoted as $\mathbf{I} \in \mathbb{R}^{C\times H\times W}$, with H, W, and C indicating height, width, and number of channels respectively. This module generates hierarchical features from the input image, which are essential for predicting the positions and types of vertices and for delineating instance-level segments, either through edge detection or centerline tracing. The UT-Hub acts as the receiver, processing the predicted segments and keypoints. It utilizes a graph segments transformer to define boundaries or centerlines and a keypoint regression transformer to precisely locate keypoints. Lastly, TMX manages the topological connections and iteratively predicts subsequent keypoints based on its stored historical graph.

2.2 Universal-Topology Hub

In order to construct a coherent graph representation within a unified space, UT-Hub transforms image tokens into two specialized branches: graph segment prediction and keypoint regression. The pathway dedicated to graph segment prediction aims to accurately forecast segments that coincide with boundaries or centerlines. Concurrently, the keypoint regression pathway is tailored to meticulously estimate the positions of keypoints. This bifurcated approach ensures that both the structural and positional aspects of the graph are captured and integrated effectively, facilitating a robust and dynamic representation of agent topology during the graph updating phase.

Graph segments prediction. Upon processing the cropped input image with a feature extractor, we obtain hierarchical features denoted by $\boldsymbol{f_1}, \boldsymbol{f_2}, \ldots, \boldsymbol{f_n}$. These features, characterized by varying dimensions, are subsequently relayed to the decoder. The resultant combined feature map, $\boldsymbol{\mathcal{F}}$, is synthesized as $\boldsymbol{\mathcal{F}} = \boldsymbol{f_1} \oplus \boldsymbol{f_2} \oplus \ldots \oplus \boldsymbol{f_n}$, where $\oplus$ symbolizes the fusion operation applied to amalgamate the hierarchical features. This composite map, $\boldsymbol{\mathcal{F}}$, is then transformed into image tokens through a flattening operation. Further refinement is achieved by appending positional encoding and an *cls* token to these flattened tokens. They are then fed into a multi-head self-attention mechanism within the transformer architecture. The transformer's output yields a probability distribution, denoted by $\mathcal{P} = \{P_1, P_2, ..., P_N\}$, for the recovery of multi-class segments, where P_i signifies the probability associated with segment S_i. The training process is informed by the ground truth segmentation, utilizing a softmax cross-entropy loss function to facilitate multi-class learning, formalized as follows:

$$\mathcal{L}_{seg} = -\frac{1}{C}\sum_{i=0}^{C} y_i \cdot \log(y_{pi}), \tag{1}$$

where y_i represents the label of the ith class, and $y_{pi} = \frac{e^{x_i}}{\sum_{j=0}^{N} e^{x_j}}$ delineates the predicted probability map, with C denoting the total number of classes.

Keypoint Regression. Concurrently with graph segments prediction, our approach also includes the prediction of keypoints, which are critical for estimating subsequent keypoints within the TMX module. This process diverges from the typical sequence-to-sequence translation by directly regressing the coordinates of each keypoint, employing a binary cross-entropy loss function:

$$\mathcal{L}_p = \text{BCELoss}(\bar{\mathcal{V}}, \hat{\mathcal{V}}), \tag{2}$$

where $\bar{\mathcal{V}} = \{\bar{v}_i\}_{i=1}^{M}$ signifies the set of predicted keypoints and $\hat{\mathcal{V}} = \{\hat{v}_i\}_{i=1}^{M}$ corresponds to the set of ground truth keypoints.

As each iteration of the TMX module dynamically updates the historical graph, it is important to elucidate how keypoint regression for the forthcoming step is computed. Consider the scenario at step $t+1$, where the predicted keypoints are $\{\bar{v}_i\}_{i=1}^{K}$ and the ground truth keypoints are $\{\hat{v}_i\}_{i=1}^{M}$. To identify the best alignment between predicted and actual keypoints, we undertake an optimization process. This involves solving an assignment problem, commonly addressed through a matching algorithm (Date and Nagi, 2016) that pairs each predicted keypoint with a ground truth counterpart in the most efficient manner:

$$\hat{\zeta} = \underset{\zeta}{\text{argmin}} \sum_{i}^{K} \mathcal{M}(\bar{v}_i, \hat{v}_\zeta), \tag{3}$$

where ζ is the indicator that maps the predicted keypoints to their ground truth counterparts, minimizing the pairwise Euclidean distances as captured by $\mathcal{M}(.)$ matching function. Once the vertices have been matched, the discrepancy between the coordinates is quantified using an L1 loss:

$$\mathcal{L}_\zeta = \frac{1}{M}\sum_{i}^{M} |\bar{v}_\zeta - \hat{v}_i|. \tag{4}$$

It allows for a precise alignment of the predicted keypoints with the actual graph structure, maintaining consistency and precision across iterations.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-3-2024-445-2024 | © Author(s) 2024. CC BY 4.0 License.

Joint Learning. For each predicted keypoint $\bar{v}_i$, its associated probability is obtained from Equation 2. During the graph updating phase, we retain only the keypoint with the highest probability from the predicted set that corresponds to the ground truth point $\hat{v}_\zeta$. Specifically, if multiple predicted keypoints match $\hat{v}_\zeta$, only those with the highest scores are selected. The overall loss function for joint learning of graph segmentation and keypoint detection combines the individual loss functions:

$$\mathcal{L} = \mathcal{L}_{seg} + \alpha\mathcal{L}_p + \beta\mathcal{L}_\zeta, \tag{5}$$

where the coefficients α and β are used to balance the contributions of each component to the total loss.

2.3 Topology Memory Mixer

The determination to next valid keypoint and topological graph updating is illustrated in Alg. 1. After making predictions with UT-Hub, we identify K valid vertices adjacent to v_t, forming the set $\mathcal{V}^K$=$\{v_{t+1}^i\}_{i=1}^K$. Non-Maximum Suppression (**NMS**) is then applied to $\mathcal{V}^K$ to derive the required vertex set $\mathcal{V}_p$. From this set, the vertex indexed by σ with the highest score is chosen as the initial keypoint for the current graph $\mathcal{G}_t$ at step t. Subsequently, we compute the distance d_t between this initial vertex v_t and its adjacent vertex v_{t+1}^i, and calculate the probability P_t for this vertice.

Algorithm 1: Topology Memory Mixer

Input: Cropped Image Region $\mathbf{I}$ and candidate agent vertices $\mathcal{V}^K = \{v_{t+1}^i\}_{i=1}^K$
Output: Topology graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

begin
  $\mathcal{V}^p \leftarrow \mathbf{NMS}(\mathbf{I}, \mathcal{V}^K)$
  $t \leftarrow 0$
  $\sigma \leftarrow \arg\max P(\mathcal{V}^p)$
  $\mathcal{G}_t \leftarrow v_\sigma$
  while $\mathcal{V}^p$ *not empty* **do**
    $d_t \leftarrow \left\|v_t - v_{t+1}^i\right\|^2$
    $P_t \leftarrow P(v_{t+1}^i)$
    if $|\mathcal{V}^p| = 0$ **then**
      break
    else if $|\mathcal{V}^p| = 1$ *and* $d_t \geq T_1$ *and* $P_t \geq T_2$ **then**
      $v_t \leftarrow \bar{v}_{t+1}^1$
      $t \leftarrow t + 1$
      $\mathcal{V}^p = \mathcal{V}^p \backslash \mathcal{V}^p.pop()$
      $\mathcal{G} \leftarrow \mathcal{G}_t \cup v_t$
      break
    else if $|\mathcal{V}^p| > 1$ **then**
      $\mathcal{G}_t \leftarrow \mathcal{G}_t \cup \mathcal{V}^p.pop()$
      $\mathcal{G} \leftarrow \mathcal{G}_t$
  end
  return $\mathcal{G}$
end

If $\mathcal{V}_p$ is empty, we perform **NMS** on another cropped image region. However, if $\mathcal{V}p$ contains only one vertex and it satisfies both criteria — distance d_t exceeds threshold T_1 and probability P_t surpasses threshold T_2 — this vertex, $\bar{v}_{t+1}^1 \in \mathcal{V}_p$, is chosen as the final addition to the graph $\mathcal{G}$, halting any further updates to $\mathcal{G}_t$. On the other hand, if $\mathcal{V}_p$ includes multiple vertices, each is evaluated through Maximum Posterior Estimation (MAP). Vertices that meet the required standards are integrated into $\mathcal{G}_t$, and these changes are subsequently incorporated into the main graph $\mathcal{G}$, ensuring it accurately reflects the historical graph. We repeat the above process until there exists no element in $\mathcal{V}_p$.

In fact, in the processing of subsequent vertices v_{t+1}^i, we continuously conduct Maximum a Posteriori (MAP) estimation until the optimal vertex is identified. The UT-Hub takes K query vertex queries $Q = \{q_i\}_{i=1}^K$ and predicts K adjacent vertices. We define the achieved vertices and its associated probility as the reward:

$$\mathcal{V}^K = \{\underset{v_{t+1}^i}{\operatorname{argmax}} P[v_{t+1}^i | \text{UT-Hub}(Q)]\}_{i=1}^K, \tag{6}$$

Each 2D vertex v_{t+1}^i is decoded and assigned a validity probability $P(v_{t+1}^i)$. This probability quantifies the likelihood that v_{t+1}^i is considered valid for integration into the graph G and is served as the reward score.

3. EXPERIMENTS

3.1 Datasets and Implementation Details

TOPO-BOUNDARY (Xu et al., 2021b) is an extensive collection tailored for enhancing road curb detection through aerial imagery, comprising 25,295 four-channel images each sized at 1000×1000 pixels. Accompanying each image are eight training labels for various sub-tasks. We have partitioned these images into distinct subsets: 10,236 patches for training, 1,770 patches for validation, and 3,289 patches for testing. This structured dataset is pivotal for topological graph extraction, where accurate road boundary delineation derived from detailed curb detection.

REALSCENE is a specialized urban scene remote sensing collection that aligns high-resolution three-channel satellite images, sized at 1000×1000 pixels, with their corresponding Open Street Map (OSM) data, totaling 2685 images.This dataset is particularly valuable for its inclusion of vectorized topological graphs of buildings and roads, providing rich details necessary for advanced urban mapping and analysis. We have randomly shuffled this dataset and partitioned it into training, validation, and testing subsets, which constitute 70%, 20%, and 10% of the total dataset, respectively.

Implementation details. Our models are trained on eight Tesla V100 GPUs with an initial learning rate (LR) of $1e^{-5}$, adjusted over 80 epochs using a poly strategy with a power of 0.95. The first 30 epochs feature a warm-up phase, reducing the LR to $0.1\times$ its initial value. We utilize the AdamW optimizer, celebrated for its efficiency, with settings including an epsilon of $1e^{-8}$, weight decay of $1e^{-2}$, and a batch size of 2 per GPU. Augmentation techniques such as random resizing (ratio 0.5–2.0), color jitter, horizontal flipping, and Gaussian blur are employed to bolster model robustness. Additionally, ImageNet-1K pre-trained weights are integrated into the accompanying branch to enhance performance comparability.

Evaluation metrics. Unlike previous metrics that primarily concentrate on overall pixel-level metrics, our approach emphasizes evaluating the boundary status and the corresponding topological graph within a trimap. In alignment with common practices, we adopt precision, recall, F1-score, averaged path length similarity (APLS), and mean intersection over union (mIoU) as our metrics, applying them within a trimap of three pixels.

				TOPO-BOUNDARY					REALSCENE				
	Model name	**#Params(M)**	**GFLOPs**	**Precision@3**	**Recall**	**F1**	**APLS**	**mIoU**	**Precision@3**	**Recall**	**F1**	**APLS**	**mIoU**
Segmentation-based	FCN	54.32	204.83	61.4	63.8	62.4	5.7	56.3	56.3	56.6	56.4	0.4	18.7
	OCRNet	62.06	806.80	58.7	61.9	60.1	7.9	54.6	54.8	50.9	50.6	1.1	48.0
	SegFormer	61.94	161.79	59.9	63.0	61.2	9.3	55.4	53.8	55.0	54.2	1.3	48.2
	BEiT	360.72	3727.34	59.2	62.5	60.5	7.5	55.0	54.8	56.6	55.5	0.9	48.9
Graph-based	RoadTracer	27.30	204.79	68.9	66.6	67.7	14.2	60.2	50.4	50.5	50.4	0.1	43.0
	VecRoad	31.69	504.65	67.8	54.6	57.2	7.3	53.3	49.7	48.9	28.8	2.7	18.1
	ConvBoundary	46.95	334.61	72.8	63.1	66.5	14.1	59.4	49.0	46.4	44.5	1.1	36.4
	DAGMapper	31.56	213.41	68.9	65.5	67.0	18.1	59.7	52.4	50.2	7.2	1.3	3.8
	Enhanced-iCurb	46.07	253.01	50.3	50.2	50.2	2.4	49.2	50.8	51.5	22.3	1.3	12.7
	TopDiG	23.56	448.74	75.4	69.3	71.1	16.5	62.7	67.3	50.2	44.7	1.6	39.9
Contour-based	PolyWorld	39.53	2054.81	-	-	-	-	-	53.1	52.4	52.7	1.3	48.2
	Frame Field	117.14	1851.74	-	-	-	-	-	51.4	50.9	51.0	1.5	48.3
	HiSup	74.29	725.74	-	-	-	-	-	54.2	51.5	51.8	1.4	49.0
Agent-based	TopScene	124.33	1883.44	78.3	62.5	69.5	18.3	65.26	82.8	43.8	57.3	4.6	53.2

Table 1. Evaluation on TOPO-BOUNDARY and REALSCENE. We count the number of parameters (#Params) and GFLOPs with the size of 640×640 pixels. Precision@3 indicates the score is evaluated with a buffer size of three pixels.

3.2 Comparison against the State of the Art

To verify the efficacy of TopoSense framework, we conduct extensive experiments on these two datasets. And the ablation studies are conducted on REALSCENE dataset. Table 1 shows the comparisons against the state-of-the-arts.

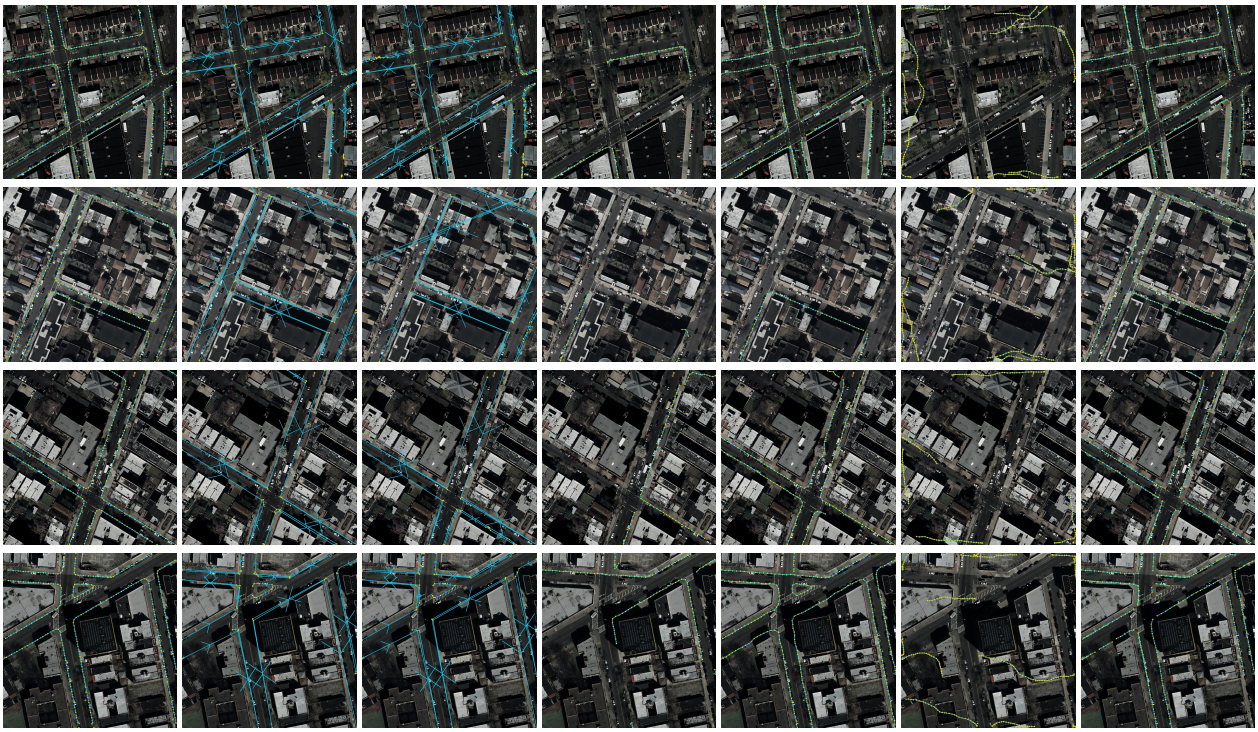
Results on TOP-BOUNDARY. Table 1 presents a comprehensive comparison of our TopoSense against other recent methods on the TOP-BOUNDARY dataset. Overall, TopoSense achieves state-of-the-art performance across most of the metrics. Notably, the APLS score is marginally lower in segmentation-based methods, ranging from 5.7% to 9.3%. This suggests that relying solely on the Douglas–Peucker simplification as post-processing may negatively affect the topological structure. In contrast, graph-based methods better preserve the topological integrity, thereby enhancing the APLS status. However, these methods still suffer from low recall rates when addressing complex road curbs. Our approach maintains the topological graph and strikes a balance between preserving the topological integrity and managing intricate structures. In the case of contour-based algorithms, direct application to linear boundaries, such as road curbs, leads to detrimental effects since they are not dedicated design for the open-set curbs. Nonetheless, TopoSense overcomes this limitation with minimal topological loss, outperforming segmentation-based, graph-based, and contour-based baselines by significant margins of $+9.10\%$ in APLS.

Results on REALSCENE. The results in the fourth column of Table 1 highlight challenges in the REALSCENE dataset, which features smaller polygon-shaped buildings and polyline-shaped roads. Most methodologies reviewed failed to significantly enhance performance, especially in boundary or centerline accuracy. The Average Path Length Similarity (APLS) scores remained below 5%. This limitation likely stems from two factors. First, the training samples exhibit wide scale variability, from tiny buildings to extensive polylines. Second, the different encoding strategies for polylines and polygons may impede the model's ability to provide consistent boundary sequence lengths. Consequently, adjusting the number of keypoints offers minimal benefit. Despite these obstacles, our TopoSense model stands out, achieving a 30% increase in precision over the segmentation-based baseline. It also surpasses graph-based methods, with a 10% improvement in F1 scores and a 4% increase in mIoU scores. When handling topological graphs, TopoSense demonstrates significant enhancements: it shows gains of 17.52% mIoU compared to the BEiT. Moreover, TopoSense operates with roughly one-third the parameters of the BEiT method. These results underscore TopoSense's superior adaptability and efficiency in processing complex spatial datasets, suggesting its strong potential for precision-focused applications.

Segmentation-based VS. Graph-based. As can be seen from Table 1, the segmentation-based approaches like FCN and OCRNet show robustness on the TOPO-BOUNDARY dataset with high Recall and F1 scores, indicating their effectiveness in region-based accuracy. However, these methods demand considerable computational resources, exemplified by SegFormer's GFLOPs reaching 161.79, potentially limiting their usage in the resource-constrained environments. On the REALSCENE, segmentation-based methods fall behind, with lower Precision@3 and mIoU scores than their graph-based counterparts, signaling difficulties in capturing intricate boundary or centerline topologies. In contrast, graph-based models such as RoadTracer and DAGMapper, while less computationally efficient with GFLOPs—for instance, VecRoad at 504.65—exhibit superior precision, particularly in the Precision@3 metric on TOPO-BOUNDARY. However, their performance in terms of Recall and F1 is modest compared to the segmentation-based methods on the same dataset. Notably, graph-based approaches outshine with higher APLS scores on REALSCENE, suggesting an advantage in capturing the likeness of boundary or centerline paths, despite lower region-based accuracy as indicated by mIoU scores. While agent-based TopoSense take the advantages of segmentation-based method for keypoints predication and the local graph encoding in graph-based approaches.

Contour-based VS. Graph-based. The contour-based method, Frame Field, displays moderate Precision@3 and F1 scores of 51.4% and 51.0%, respectively, on the REALSCENE dataset. These scores suggest a competitive edge in scenarios requiring precise edge delineation. However, with a substantially high computational cost, as indicated by its GFLOPs of 1851.74, contour-based methods may be less viable for resource-sensitive applications. Additionally, they exhibit a lower mIoU score, at 48.3% on REALSCENE, indicating a potential shortfall in region-based accuracy compared to graph-based approaches. Graph-based methods, on the other hand, demonstrate a more balanced performance with respect to precision and computational load. For example, TopoDiG achieves a higher mIoU score of 62.7% on TOPO-BOUNDARY with comparatively lower GFLOPs at 448.74, illustrating its efficiency and accuracy in delineating complex structures. However, these methods do not universally excel, as seen with Enhanced-iCurb's APLS score of 2.4% on TOPO-BOUNDARY, suggesting possible improvements in path similarity metrics. By contrast, TopoSense retains a balance of achieving competitive scores and computational efficiency.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-3-2024-445-2024 | © Author(s) 2024. CC BY 4.0 License.



(a) Ground truth (b) SegFormer (c) BEiT (d) VecRoad (e) DAGMapper (f) Enhanced-iCurb (g) Ours

Figure 4. Qualitative demonstrations on TOPO-BOUNDARY. We visualize the road curb extraction results with line-shaped targets. The size of each image is 1000×1000. (a) Ground-truth graph (cyan lines). (b)-(c) The road network graph predicted by segmentation-based approaches (cyan lines). These two approaches have poor topology performance such as incorrect disconnections. (d)-(f) The road network graph predicted by graph-based approaches (we represent the topological orientation with arrows, best view by zooming in). Compared with VecRoad, DAGMapper, Enhanced-iCurb does not produce desired outputs due to the reliance of initial vertices that are diffult to train.

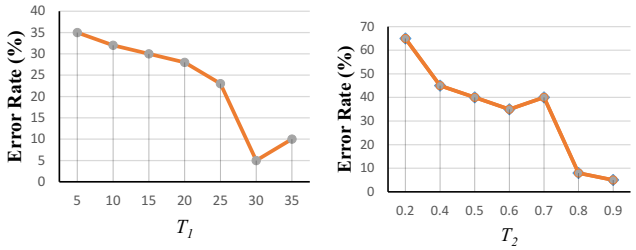


Figure 5. Error rate with respect to T_1 and T_2

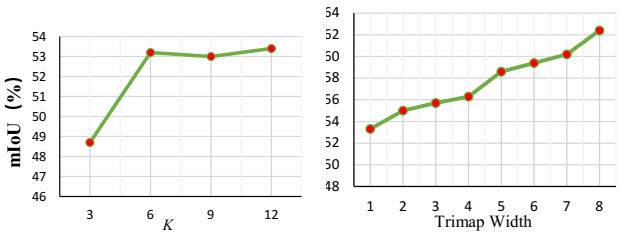


Figure 6. Impact of K and trimap width.

3.3 Ablation Studies on REALSCENE

Analysis of threshold T_1 and T_2. Both thresholds T_1 and T_2 are crucial in determining the agent's subsequent status in our model. We utilize probability estimates derived from Maximum a Posteriori (MAP) estimation techniques. As depicted in Figure 5, we systematically evaluate the impact of varying distances $\{5, 10, 15, 25, 30, 35\}$ on error rates. The minimum error rate occurs at a distance of 30 units, with a quantified error reduction of 18% compared to the next best setting at 25 units. This suggests that T_1 is optimally set around this distance to minimize predictive inaccuracies. Regarding T_2, which regulates the number of vertices selected for further processing, our analysis indicates a direct correlation between T_2 and vertex selection: higher T_2 values correspond to stricter selection criteria, effectively reducing the number of points chosen. Quantitatively, increasing T_2 from a lower threshold to an upper threshold results in a selection reduction of approximately 40%, emphasizing its sensitivity and the need for careful calibration.

Analysis of query vertex number K. As depicted in Figure 6, we conducted a comprehensive examination of the impact that varying the number of query vertices has on the boundary IoU score. The results indicate that performance peaks when $K = 6$. As K increases beyond this value, performance gains diminish, culminating in a plateau at $K = 10$. This plateau likely results from the dynamic adjustments made by the UT-Hub, which is influenced by the regressed ground truth points. Consequently, performance fluctuations remain minimal, within a range of $\pm 0.5\%$ mIoU. This observed stabilization implies that the system reaches a point of efficiency, effectively balancing computational load and accuracy. A closer quantitative analysis shows that incrementing K from 6 to 10 results in a marginal increase in mIoU by only 0.2%, demonstrating the principle of diminishing returns as query vertices increase. These findings elucidate a critical threshold for K, suggesting that exceeding this number does not proportionally enhance performance and instead leads to unnecessary computational expenditure.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-3-2024-445-2024 | © Author(s) 2024. CC BY 4.0 License.

Structure	#Params(M)	GFLOPs	mIoU(%)
TopSense	124.33	1883.44	53.2
– with ResNet50	205.58	687.78	52.6 (-0.6)
– without ResNet101	319.22	775.05	53.4 (+0.2)
– with ConvNeXT	2153.00	2023.92	53.3 (+0.1)
– without Swin-T	224.83	740.89	55.1 (+1.8)
– without Swin-B	580.94	1148.00	55.3 (+2.1)
– without Swin-L	1232.00	1845.92	54.9 (+1.7)

Table 2. Ablation study of the TopSense backbones.

Impact of different trimap width. As shown in Figure 6, we investage the effects of different number of trimap width for topological graph extraction. The incremental increase in mIoU with respect to trimmap width observed in this figure has significant implications for topological graph extraction. The positive correlation between trimap width and mIoU suggests that providing a wider contextual margin allows the agents to better distinguish between the features of interest and the surrounding environment. The incremental increase in mIoU with the widening of the trimap width from 1 to 8 suggests that the agents benefit from a broader spatial context when identifying and delineating topological features. A trimap width that is too narrow may restrict the agent's perception, impeding its ability to accurately resolve complex spatial relationships for topological graph extraction. As the trimmap width expands, the agents appear to gain a more comprehensive view of the surroundings, likely aiding in more precise decision-making processes for edge detection and graph construction. The consistent upward trend observed in the graph indicates that a wider trimap allows for better distinction between the pertinent features and the background.

Impact of different backbone. Table 2 presents a comparative analysis of the TopSense structure's performance when integrated with various backbone architectures versus its performance when those backbones are omitted, as measured by the number of parameters, computational cost (GFLOPs), and mIoU. The baseline TopSense model, without any additional backbones, is relatively lightweight with 124.33 million parameters and requires 1883.44 GFLOPs for processing, achieving a 53.2% mIoU. When enhanced with ResNet50, the parameter count increases to 205.58 million and the computational cost significantly drops to 687.78 GFLOPs, a sign of increased efficiency. However, this integration slightly reduces the mIoU by 0.6%. The exclusion of ResNet101 yields a slightly heavier model with 319.22 million parameters and a lower computational cost of 775.05 GFLOPs, while marginally increasing mIoU by 0.2%. The integration with ConvNeXT leads to a substantial increase in parameters to 2153.00 million and GFLOPs to 2023.92, with a marginal mIoU gain of 0.1%. Omitting Swin Transformer variants shows a varied increase in mIoU: Swin-T, Swin-B, and Swin-L contribute to gains of 1.8%, 2.1%, and 1.7% respectively. Notably, these omissions also reduce the computational cost to varying degrees, with Swin-T requiring 740.89 GFLOPs, Swin-B at 1148.00 GFLOPs, and Swin-L at 1845.92 GFLOPs. The parameters also reduce correspondingly with the exclusion of these backbones.

Impact of agent's reward. We use the MAP to estimate the probality for the next vertices that current agent (vertice) is to be connected. This posteriori estimation is served as the reward function for the agent. The impact factor for this reward is the total layers of transformer blocks in UT-Hub. We further analysize such factors with different layers of Transformer blocks, which offers a quantitative insight into the impact of transformer block architecture on the learning efficacy in an agent-driven topological graph extraction framework. A discernible trend of enhanced performance is observed with an increasing count of transformer blocks within the model's architecture. Models comprising 12, 24, and 36 transformer blocks exhibit a concomitant rise in mIoU values across training epochs, with the 36-block model achieving an mIoU of nearly 60%, juxtaposed against approximately 50% for the 12-block model. This gradation of performance, showcased over 80 epochs, substantiates the pivotal role of architectural depth, particularly emphasizing that models with more transformer blocks are more adept for agent-driven tasks, a trait crucial for accurate vertex connectivity in the realm of topological graph extraction. The ascent in mIoU is especially steep in the early epochs, indicating a rapid initial learning phase, followed by a plateau that suggests a diminishing return on model complexity as the models approach their learning capacity.

4. Conclusion

In conclusion, TopoSense represents a paradigm shift in the extraction of topological graphs from remote sensing images, marking a substantial leap forward in the field of geospatial data analysis. This innovative framework, by harnessing the synergistic potential of reinforcement learning and advanced neural network architectures, stands out by autonomously navigating the intricate maze of pixel-level data. It achieves not only a higher degree of accuracy in spatial feature detection but also a remarkable reduction in processing time. The comprehensive experiments conducted on REALSCENE and the TOP-BOUNDARY dataset lay testament to the superior capabilities of TopoSense. It adeptly captures complex spatial relationships, delineating a clear advance over conventional methodologies that are often constrained by scalability and adaptability issues. As the demand for accurate and efficient topological graph extraction grows, TopoSense offers an adaptable and scalable solution that is well-poised to meet the challenges presented by the increasing complexity of remote sensing data. It embodies the potential to revolutionize the automation of cartographic representation and could serve as the cornerstone for future innovations in remote sensing image interpretation.

Limitations and future work. Agent-driven topological graph extraction from remote sensing images is poised to be significantly advanced by leveraging the power of language models and the push towards more efficient, lightweight computational architectures. The integration of these elements promises to catalyze a new era of high-performance, scalable remote sensing technologies that can operate across various agent types and applications. Language models, particularly those fine-tuned for spatial data interpretation, could be pivotal in improving the semantic understanding of remote sensing images, enabling agents to discern complex features and relationships with greater precision. When coupled with lightweight neural networks, these agents can perform tasks with reduced computational overhead, making them more accessible for deployment on edge devices and in real-time systems. Furthermore, multi-agent collaboration, powered by advancements in communication protocols and decentralized learning, can amplify efficiency, whereby different agents specialize in distinct aspects of the task, such as feature detection, boundary delineation, or object classification. Such collaboration can lead to a more nuanced extraction of topological graphs, effectively addressing the diverse and dynamic nature of spatial structures in remote sensing images.

5. Acknowledgment

This work was supported by the Key Research and Development Program of Hubei Province (No. 2023BAB173), funded by State Key Laboratory of Geo-Information Engineering, NO. SKLGIE2021-M-3-1, and was supported in part by the Special Fund of Hubei Luojia Laboratory (No. 220100028).

References

Bastani, F., He, S., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., Madden, S., DeWitt, D., 2018. Roadtracer: Automatic extraction of road networks from aerial images. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4720–4728.

Belli, D., Kipf, T., 2019. Image-conditioned graph generation for road network extraction. *arXiv preprint arXiv:1910.14388*.

Chen, S., Zhang, Y., Liao, B., Xie, J., Cheng, T., Sui, W., Zhang, Q., Huang, C., Liu, W., Wang, X., 2023. Vma: Divide-and-conquer vectorized map annotation system for large-scale driving scene. *arXiv preprint arXiv:2304.09807*.

Date, K., Nagi, R., 2016. GPU-accelerated Hungarian algorithms for the linear assignment problem. *Parallel Computing*, 57, 52–72.

Douglas, D. H., Peucker, T. K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2), 112–122.

Girard, N., Smirnov, D., Solomon, J., Tarabalka, Y., 2021. Polygonal building extraction by frame field learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5891–5900.

Hatamizadeh, A., Sengupta, D., Terzopoulos, D., 2020. End-to-end trainable deep active contour models for automated image segmentation: Delineating buildings in aerial imagery. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, Springer, 730–746.

Lian, R., Huang, L., 2020. DeepWindow: Sliding window based on deep learning for road extraction from remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 1905–1916.

Ling, H., Gao, J., Kar, A., Chen, W., Fidler, S., 2019. Fast interactive object annotation with curve-gcn. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5257–5266.

Liu, J., Ding, H., Cai, Z., Zhang, Y., Satzoda, R. K., Mahadevan, V., Manmatha, R., 2023a. Polyformer: Referring image segmentation as sequential polygon generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18653–18663.

Liu, Y., Yuan, T., Wang, Y., Wang, Y., Zhao, H., 2023b. Vectormapnet: End-to-end vectorized hd map learning. *International Conference on Machine Learning*, PMLR, 22352–22369.

Peng, S., Jiang, W., Pi, H., Li, X., Bao, H., Zhou, X., 2020. Deep snake for real-time instance segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8533–8542.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wei, S., Ji, S., Lu, M., 2019. Toward automatic building footprint delineation from aerial images using CNN and regularization. *IEEE Transactions on Geoscience and Remote Sensing*, 58(3), 2178–2189.

Wei, S., Zhang, T., Ji, S., Luo, M., Gong, J., 2023. BuildMapper: A fully learnable framework for vectorized building contour extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 197, 87–104.

Wei, Y., Hu, X., Zhang, M., Xu, Y., 2020. Automatic extraction of road centerlines and edge lines from aerial images via CNN-based regression. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 2, 925–932.

Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., Shen, C., Luo, P., 2020. Polarmask: Single shot instance segmentation with polar representation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12193–12202.

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

Xu, Z., Liu, Y., Sun, Y., Liu, M., Wang, L., 2023. Rngdet++: Road network graph detection by transformer with instance segmentation and multi-scale features enhancement. *IEEE Robotics and Automation Letters*.

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

Xu, Z., Sun, Y., Liu, M., 2021b. Topo-boundary: A benchmark dataset on topological road-boundary detection using aerial images for autonomous driving. *IEEE Robotics and Automation Letters*, 6(4), 7248–7255.

Yan, X., Ai, T., Yang, M., Tong, X., 2021. Graph convolutional autoencoder model for the shape coding and cognition of buildings in maps. *International Journal of Geographical Information Science*, 35(3), 490–512.

Zhu, C., Zhang, X., Li, Y., Qiu, L., Han, K., Han, X., 2022. Sharpcontour: a contour-based boundary refinement approach for efficient and accurate instance segmentation. *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 4392–4401.

Zhu, Y., Huang, B., Gao, J., Huang, E., Chen, H., 2021. Adaptive polygon generation algorithm for automatic building extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1–14.

Zorzi, S., Bittner, K., Fraundorfer, F., 2020. Map-repair: Deep cadastre maps alignment and temporal inconsistencies fix in satellite images. *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 1829–1832.