ISPRS Journal of Photogrammetry and Remote Sensing 209 (2024) 213–232

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

From lines to Polygons: Polygonal building contour extraction from High-Resolution remote sensing imagery

Shiqing Wei [a,b], Tao Zhang [b], Dawen Yu [b], Shunping Ji [b,*], Yongjun Zhang [b], Jianya Gong [b]

[a] College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao 266580, China
[b] School of Remote Sensing and Information Engineering, Wuhan University, 129 Luoyu Road, Wuhan 430079, China

ARTICLE INFO

Keywords:
Building extraction
Feature line detection
Transformer
Topological reconstruction
Remote sensing images

ABSTRACT

Automated extraction of polygonal building contours from high-resolution remote sensing images is important for various applications. However, it remains a difficult task to achieve automated extraction of polygonal buildings at the level of human delineation due to diverse building structures and imperfect image conditions. In this paper, we propose Line2Poly, an end-to-end approach that uses feature lines as geometric primitives to achieve polygonal building extraction by recovering topological relationships among these lines within an individual building. To extract building feature lines with precision, we adopt a two-stage strategy that combines Convolutional Neural Network (CNN) and transformer architectures. A CNN-based module extracts preliminary feature lines, which serve as positional priors for initializing positional queries in the subsequent transformer-based module. For polygonal building contour reconstruction, we devise a learnable polygon topology reconstruction module that predicts adjacency relationships among discrete lines, and integrates lines into building polygons. The resultant building polygons, based on feature lines, exhibit inherent regularity that aligns with manual labeling standards. Extensive experiments on the Vectorizing World Buildings dataset, the WHU aerial building dataset and the WHU-Mix (vector) dataset validate Line2Poly's impressive performance in building feature line extraction and instance-level building detection. Moreover, Line2Poly's predictions exhibit the highest level of concurrence with manual delineations, with over 83% agreement on the WHU aerial building test set and 68.7/59.7% on the WHU-Mix (vector) test set I and II, respectively.

1. Introduction

High-precision and large-scale polygonal building maps are extensively used in various fields, including topographic map updates, urban planning, population density estimation, and disaster management (Yeh, 1999; Boo, 2022; Lu et al., 2004). The automated extraction of building contours from high-resolution remote sensing imagery, at a level comparable to manual delineation, has the potential to significantly reduce both labor-intensive efforts and resource expenses. Despite the considerable research conducted in this area, achieving true end-to-end automation remains a challenge. Conventional methods struggle to handle the wide range of building structures, backgrounds, and diverse imaging conditions (Osher and Sethian, 1988; Chan and Vese, 2001). Recent developments in deep learning-based approaches offer promising solutions for building extraction. However, the majority of studies treat it solely as a semantic segmentation task, focusing on binary pixel classification (Ji et al., 2018; N. Nauata and Y. Furukawa, "Vectorizing World Buildings: Planar Graph Reconstruction by Primitive Detection and Relationship Inference," Cham, 2020; Chen et al., 2022; Xiao et al., 2022). In this paper, our main objective is to achieve the extraction of vectorized and regularized building contours that can effectively replace the need for manual delineation. The inspiration for this objective comes from our earlier study (Wei et al., 2019).

Digital building contours can be naturally represented as polygons enclosed by a sequence of interconnected vertices. Vertex-based methods (Xie, 2020; Ling et al., 2019; Peng et al., 2020; Liu et al., 2021) aim to directly extract polygonal buildings by regressing the coordinates of building contour vertices in an image. However, these methods often face challenges when confronted with complex scenes, such as occluded corners. In fact, the contours obtained from vertex-based methods or edge-tracked from segmentation-based methods are usually less accurate than manual delineation. To improve the quality of

* Corresponding author.
E-mail addresses: wei_sq@whu.edu.cn (S. Wei), zhang_tao@whu.edu.cn (T. Zhang), yudawen@whu.edu.cn (D. Yu), jishunping@whu.edu.cn (S. Ji), zhangyj@whu.edu.cn (Y. Zhang), Gongjy@whu.edu.cn (J. Gong).

https://doi.org/10.1016/j.isprsjprs.2024.02.001
Received 3 November 2023; Received in revised form 7 January 2024; Accepted 1 February 2024
Available online 14 February 2024

these contours, some researchers apply a series of empirical post-processing steps to transform them into regular polygons (Wei et al., 2019; Zhao et al., 2020; Zhao et al., 2018; Zorzi et al., 2020). However, the empirical post-processing may not always be effective, and the addition of post-processing steps can introduce additional complexity to the overall solution.

To mimic the process wherein a human operator sequentially traces building contour lines to create a closed polygon, we propose an innovative end-to-end approach called Line2Poly. This method utilizes lines as geometric primitives and reconstructs the topological relationships among these lines, enabling the extraction of polygonal building contours in a bottom-up fashion. The resulting building polygons exhibit inherent regularity and align with the shape characteristics of buildings. Moreover, this approach can be expanded to include the extraction of interior feature lines within building roof structures, thus catering to a wider range of application requirements beyond the capabilities of segmentation-based or vertex-based methods.

In our approach, the accurate extraction of lines is crucial for the subsequent reconstruction of polygonal buildings. To achieve this, we rely on the robust long-range information encoding capacity of neural networks, as lines are elongated in nature. While most line extraction methods (Huang et al., 2018; Zhou et al., 2019; Xue, 2020; Dai et al., 2022) are based on CNN architectures, which excel at encoding local features, they also suffer from limited kernel sizes and receptive fields. LETR (Xu et al., 2021) integrates the transformer architecture, known for its ability to capture long-range contextual information, into line extraction. However, the attention-based transformer requires substantial data for training and may experience slow convergence speed. In this study, we propose a two-stage feature line extraction strategy that combines the strengths of CNN and transformer methodologies. Our strategy includes a CNN-based preliminary feature line extraction module and a subsequent transformer-based accurate feature line extraction module. The former module identifies potential lines using building bounding boxes and corner information and filters out redundant lines to obtain preliminary lines. The latter module uses these preliminary lines as a prior cue to initialize positional queries, resulting in meticulous extraction of feature lines.

The ultimate goal of our research is the extraction of polygonal buildings. Therefore, the final step in Line2Poly involves the reconstruction of discrete lines into polygons. However, due to the imperfect nature of automatically extracted feature lines, accurately restoring the topological relationships among these lines based solely on their coordinates presents a challenge. Taking inspiration from PolyWorld (Zorzi et al., 2021), which employs neural networks to predict adjacency relations among corner points, we introduce a learnable module focused on line-based polygon topology reconstruction. This module predicts adjacency relationships among the extracted discrete lines, at the same time, it effectively addresses the challenges posed by imperfect line extraction and ultimately leads to more accurate polygon extraction.

We conduct extensive experiments on three datasets, Vectorizing World Buildings dataset (Nauata and Furukawa, 2020), WHU aerial building dataset (Ji et al., 2018) and WHU-Mix (vector) building dataset (Wei et al., 2023; Luo et al., 2208), to evaluate the performance of our proposed Line2Poly approach. The experimental results show that Line2Poly can achieve superior performance in line extraction (measured by Structural Average Precision (SAP)), instance-level building detection (measured by Average Precision (AP) and Average Recall (AR)), and manual-level building delineation (measured by the Valid Polygon Ratio (VPR)). Our primary contributions in this work are as follows:

(1) We propose Line2Poly, an end-to-end framework designed for polygonal building extraction. This innovative framework employs feature lines as fundamental geometric primitives and effectively generates building polygons through topological relationship reconstruction among the lines. Line2Poly can also be directly applied to the line extraction task.

(2) We propose a two-stage strategy for accurately extracting building feature lines. This strategy takes advantage of the strengths of convolutional neural networks (CNNs) in encoding local features and transformers in capturing long-range information. This combination results in precise extraction of building feature lines.

(3) We design a learnable module for line-to-polygon topology reconstruction. This module transforms discrete feature lines into building polygons by determining the potential adjacency relationship between contour lines.

The subsequent sections of this paper are organized as follows: Section 2 provides an overview of the related work. Section 3 presents the details of the proposed Line2Poly framework. Section 4 introduces the dataset, implementation details, and the assessment metrics employed. The evaluation results, compared with state-of-the-art methods, are presented in Section 5, along with a comprehensive discussion on the efficacy of the design choices. Finally, Section 6 encapsulates the conclusions drawn from this study.

2. Related work

In this section, we briefly review the progress of the feature line extraction methods, the development of segmentation-based polygonal building extraction methods, and the recent vertex-based and contour-based polygonal building extraction methods.

2.1. Segmentation-based polygonal building extraction

Most studies consider building extraction as a task of semantic segmentation (Ji et al., 2018; N. Nauata and Y. Furukawa, "Vectorizing World Buildings: Planar Graph Reconstruction by Primitive Detection and Relationship Inference," Cham, 2020; Yuan, 2017; Bischke et al., 2017). The emergence of fully convolutional neural network (FCN) (Long et al., 2015) and its variants such as U-Net (Ronneberger et al., October, 2015), and DeepLab (Chen et al., 2018) have significantly advanced semantic segmentation in this domain. Currently, mainstream techniques for building semantic segmentation still revolve around the FCN as the foundational framework, with specialized strategies to address issues specific to building extraction tasks (Chen et al., 2021; Zhu et al., 2020). More recently, transformer (Vaswani, 2017) have started to make their mark in computer vision. Several studies (Chen et al., 2022; Xiao et al., 2022), utilizing the Swin-Transformer (Liu, 2021) as a feature encoder, amalgamate multi-scale feature information to enhance building segmentation precision. Transformer is particularly adept at capturing global contextual information while CNN excels in extracting local features. Consequently, researchers have combined these two structures for building segmentation tasks. For example, UNetFormer (Wang, 2022) utilizes a CNN-based encoder and a transformer-based decoder to achieve precise building segmentation. Wang et al. (Wang et al., 2022) devise a dual-path feature encoding structure based on transformer and CNN. Similarly, Sun et al. (Sun et al., 2022) embrace a hybrid transformer and CNN architecture to furnish a global receptive field for each pixel. This collaborative approach maximizes the strengths of transformer and CNN, thereby augmenting the performance of semantic segmentation. Nevertheless, it should be noted that these studies focus on pixel-level segmentation. In this paper, the combination of Transformer and CNN is utilized for vectorized building contour extraction.

The segmentation maps outputted by the segmentation-based building extraction methods cannot directly obtain instance-level information. To glean building instance information, some researches (Zhao et al., 2018; Chen et al., 2023; Wu et al., 2020) have introduce instance segmentation methods, achieving simultaneous building instance localization and segmentation map extraction. However, the rasterized representations of buildings obtained by the aforementioned methods often exhibit significant jaggedness, fragmentation, and irregularities, deviating substantially from actual building shapes. The

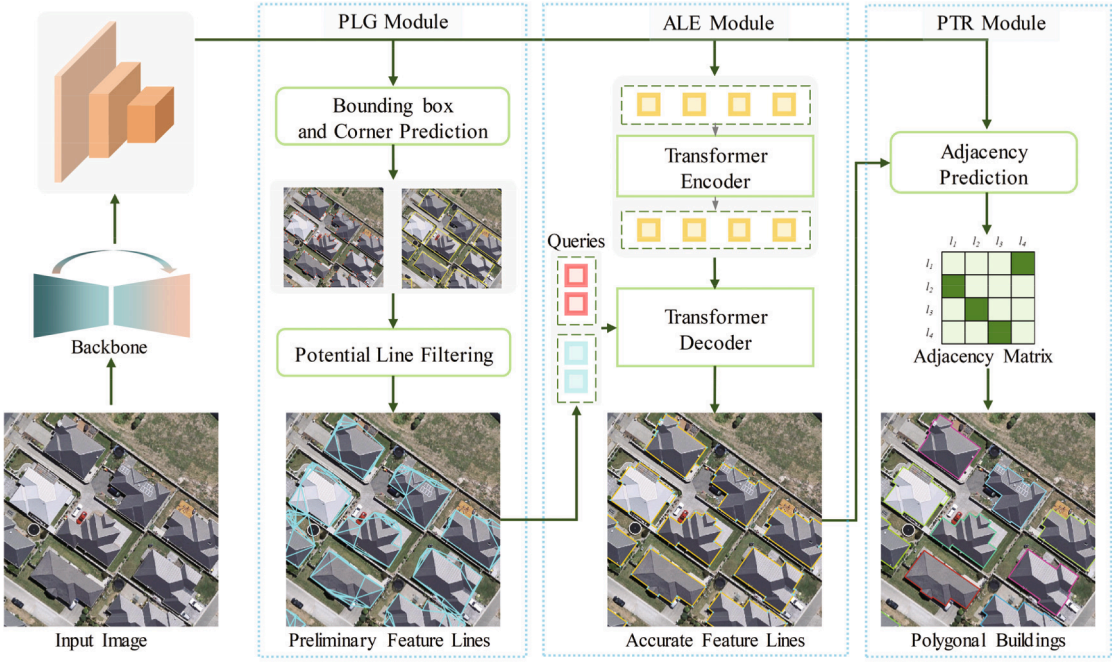

Fig. 1. The overall workflow of the proposed Line2Poly method.

simplest approach to obtain polygonal building contour representations involves transforming segmentation maps into vector formats and then employing a series of post-processing steps to refine the vector contours (Zhao et al., 2020; Zhao et al., 2018; Zorzi et al., 2020). For instance, Wei et al. (Wei et al., 2019) track building contours by identifying foreground connected domains within the segmentation map, apply a polygon simplification algorithm (the Douglas-Peucker algorithm (Douglas and Peucker, 1973), and employ empirical regularization algorithms to produce regular building polygons. Approaches like PolyTransform (Liang et al., 2020) and BOD (Chen et al., 2020) take the semantic segmentation map as initial contours and then refine it with an additional optimization module. The frame field learning method (Girard et al., 2021) introduces an extra frame field branch within semantic segmentation networks to enhance segmentation quality and facilitate polygonal representation. These methods heavily rely on the quality of building segmentation maps and typically involve multiple models or post-processing steps, resulting in complex overall workflows.

2.2. Vertex-based and contour-based polygonal building extraction

The vertex-based methods predict the corner points of buildings sequentially to obtain regular polygons. Methods like Polygon RNN (Castrejon et al., 2017) and Polygon RNN++ (Acuna et al., 2018) employ recurrent convolutional networks for polygon extraction. They initiate from an initial contour vertex and predict vertices iteratively in a predefined direction until polygon closure is achieved. This approach is well-suited for buildings with regular shapes, leading to further investigations (Huang et al., 2021; Zhao et al., 2021; Huang et al., 2021; Liu et al., 2022; Li et al., 2019) that apply and refine these techniques for polygonal building extraction. PolyWorld (Zorzi et al., 2021), on the other hand, achieves building polygon extraction by predicting building corner points and their adjacent relationships. However, this type of methods relies on the accurate extraction of building corners. For example, both Polygon RNN++ and PolyWorld often face the problem of vertex loss, resulting in incomplete building boundaries.

Recent contour-based methods for general object instance segmentation boost the building extraction study. The contour-based methods can also predict building polygons directly and have exhibited better stability than the vertex-based ones. For example, Curve GCN (Ling et al., 2019) leverages graph convolutional networks (GCN) in instance segmentation tasks. Expanding on this, TS-GCN (Wei and Ji, 2021) introduces a dual-scale approach to enhance the accuracy of building extraction. One-stage methods like PolarMask (Xie, 2020) and LSNet (Duan et al., 2104) regress object contours based on central feature information, while two-stage methods like DeepSnake (Peng et al., 2020) generates initial contours based on object positions and then optimizes these contours through neural networks. The two-stage strategy has gained increased attention (Peng et al., 2020; Liu et al., 2021; Wei et al., 2020; Zhang et al., 2022). Following this, CLP-CNN (Wei et al., 2021) and BuildMapper (Wei et al., 2023) introduce the two-stage framework to building polygon extraction, yielding favorable outcomes. To further enhance object contour quality, SharpContour (Zhu et al., 2022) introduces an effective boundary refinement module. These contour-based methods often require a substantial and redundant number of vertices to ensure the integrity of building shapes, sometimes resulting in overly smooth contours.

In contrast, taking lines as fundamental geometric primitives can offer more precise information and additional contextual cues for the learnable network. Even in the presence of occlusions, obscured areas can be reconstructed through the intersection of lines. In addition, complete building polygons can be directly produced through connecting lines sequentially without the need for additional regularization and redundant vertex removal.

2.3. Feature line extraction

Lines, a fundamental visual element in images, often play a crucial role in facilitating various downstream visual tasks (Xue, 2020), such as image matching (Xue et al., 2017) and visual SLAM (Jiang et al., 2021). The extraction of feature lines has emerged as a prominent challenge in the field of computer vision. Classical techniques for line extraction, such as the Canny operator (Canny, 1986), Hough transform (Hough, 1962) and LSD method (Von Gioi et al., 2008), have been widely used. With the advancements in neural network technology, wireframe parsing theories (Huang et al., 2018) have yielded promising results in the extraction of lines from natural images, and multiple large-scale benchmarks (Huang et al., 2018; Denis et al., 2008) have emerged. L-CNN (Zhou et al., 2019) adopts a two-stage approach, encompassing

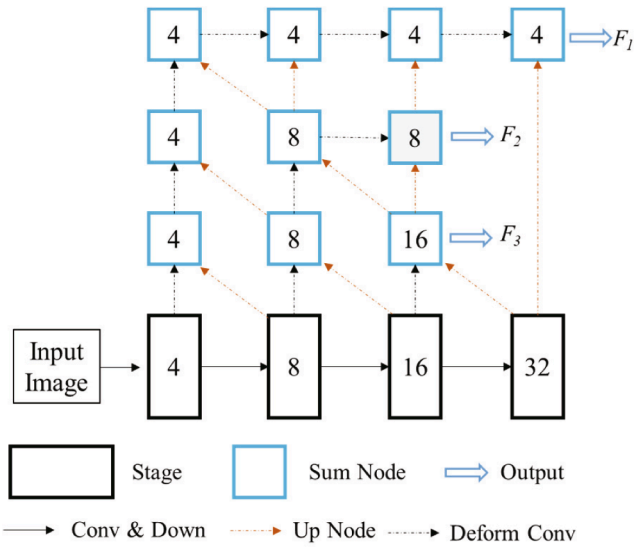

Fig. 2. The architecture of the backbone network. The numbers inside the boxes indicate the stride relative to the input image.

proposal lines generation and verification, to accomplish end-to-end line extraction. AFM (Xue et al., 2019) and HAWP (Xue, 2020) employ attractive field maps to represent lines, bridging the gap between lines and regions. Zhao et al. (Zhao et al., 2022/05/01/ 2022) follow HAWP (Xue, 2020) and incorporate a graph neural network to enhance the representation of line and vertex features. F-CLIP (Dai et al., 2022) employs a fully neural network design and parameterizes lines as center points, lengths, and angles, thus enhancing line extraction efficiency. LETR (Yuan, 2017) utilizes a transformer architecture within a two-stage model to achieve line extraction. Some studies (Nauata and Furukawa, 2020; Stekovic et al., 2021; Zhang et al., 2020) focus on the reconstruction of planar structures in buildings, which also serves the purpose of feature line extraction. Approaches like (Nauata and Furukawa, 2020; Stekovic et al., 2021) use the neural networks to extract the geometric primitives (points, lines and regions) from remote sensing images, followed by complex optimization techniques for reconstructing plane structures. Conv-MPN (Zhang et al., 2020) utilizes a message passing neural architecture to infer corners relationships and address planar graph reconstruction, but it is prone to topological errors, especially when dealing with hanging lines.

3. Methodology

The Line2Poly architecture, as shown in Fig. 1, consists of three primary modules: the Preliminary Line Generation (PLG) module, the Accurate Line Extraction (ALE) module, and the Polygons Topology Reconstruction (PTR) module. Initially, the input RGB image is fed into a shared backbone network which conducts high-level feature extraction. Then, the CNN-based PLG module predicts oriented bounding boxes (OBB) and corner points of buildings and derives potential feature lines by connecting corner points. The filtered potential lines serve as prior information for the transformer-based ALE module, facilitating the extraction of precise lines. Finally, the PTR module establishes topological interrelationships among individual building feature lines, ultimately leading to distinct and well-structured building polygons. Further details of Line2Poly are elaborated below.

3.1. Backbone network

The modified deep layer aggregation (DLA) network (Yu et al., 2018) is chosen as the foundational network for extracting high-level feature maps from the input images. These feature maps are subsequently shared with distinct sub-modules. While the original DLA network exclusively outputs 1/4 scale features, our study extends this capability to extract features at 1/8 and 1/16 scales, thus providing rich multi-scale information for subsequent modules. The architecture of the backbone network is shown in Fig. 2. Given an RGB input image $I \in R^{W\times H\times 3}$ with a width of W and a height of H, the backbone network generates feature maps at three scales, $F = \left\{F_i R^{\frac{W}{2^{i+1}}\times\frac{H}{2^{i+1}}\times(c\times i)}\right\}_{i=1}^{3}$ with $C \times i$ denoting the number of feature map channels. In this study, we set $C = 128$.

3.2. Preliminary feature lines generation

As shown in Fig. 3, the preliminary feature lines generation (PLG) module predicts the oriented bounding boxes (OBB) and corner points of buildings from the backbone feature map. By connecting the corners of a building, a set of potential lines is generated. These noisy potential lines are then subjected to a filtering process, yielding the preliminary set of feature lines. The details of OBB extraction, corner identification, potential lines generation, and subsequent filtering are provided below.

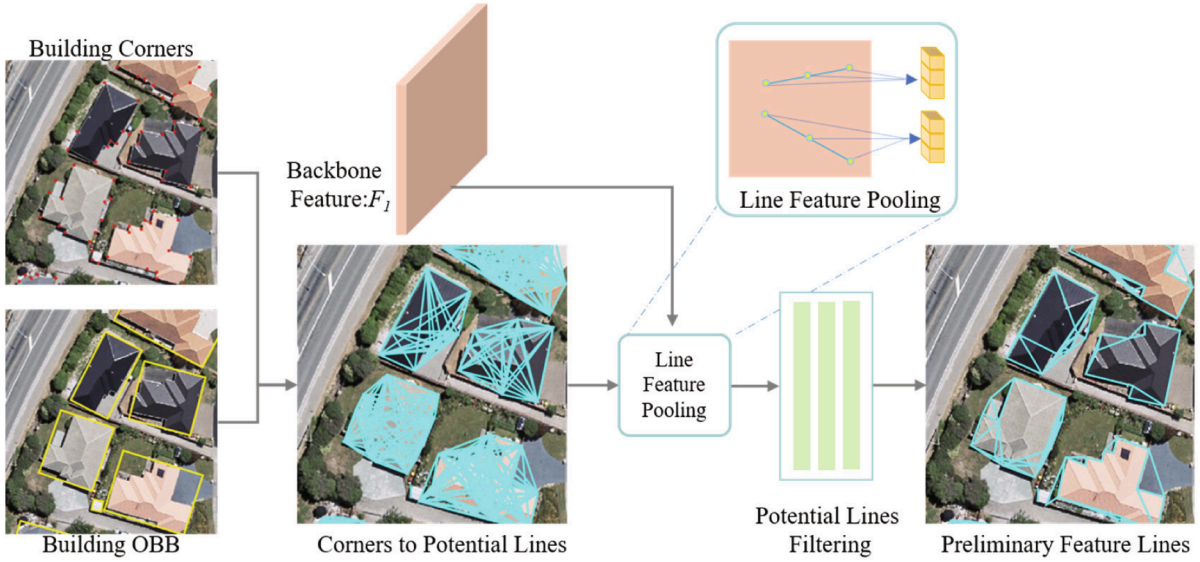

Fig. 3. Preliminary feature line generation module.

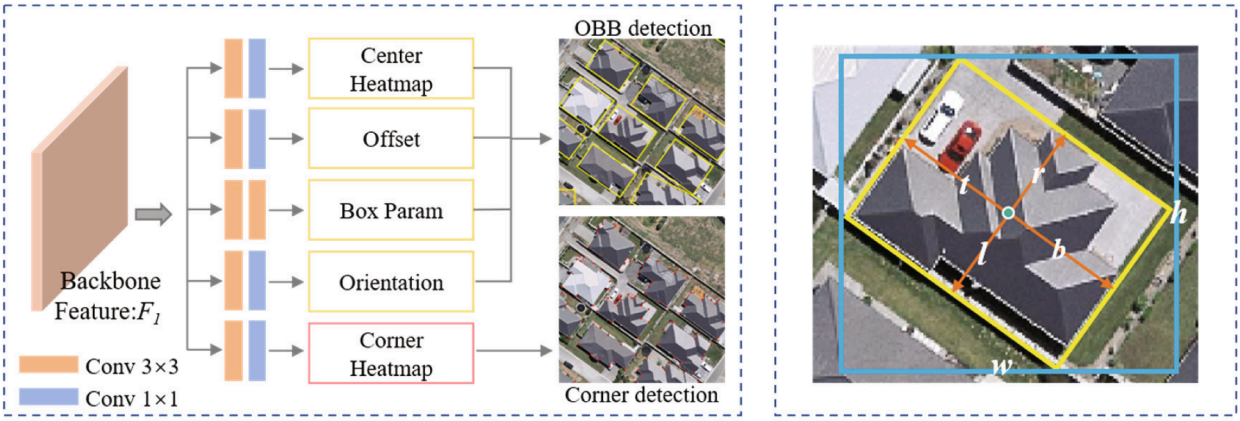

Fig. 4. Pipeline for oriented bounding boxes and corners detection in Line2Poly. Left: oriented bounding boxes and corner detection network architecture. Right: oriented bounding box parameter description based on BBAVectors method.

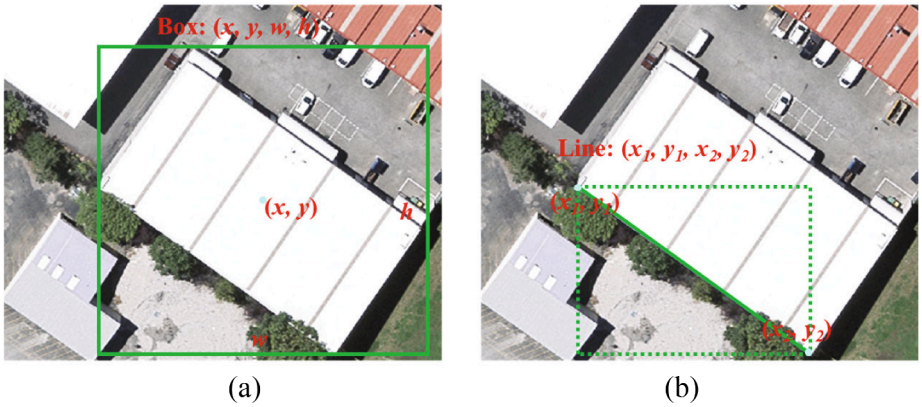

Fig. 5. Parametric representation of object detection and line extraction tasks.

3.2.1. *Oriented building bounding box extraction*

In contrast to the horizontal bounding box, OBB provides a more precise localization of the current building, thereby reducing the potential overlap with neighboring buildings. We employ the Box Boundary-Aware Vectors (BBAVectors) (Yi et al., 2021) to extracting OBB of buildings. As depicted on the left side of Fig. 4, the process of OBB extraction involves four distinct subtask branches: center heatmap prediction, offset prediction, box parameters prediction, and orientation prediction. The first two prediction branches are used to determine the building's center point, while the last two branches are responsible for generating the OBB based on the center point. These tasks require a high level of detail in image features. Therefore, only the highest-resolution feature map F1 is used as input for these purposes. For more comprehensive details, please refer to (Yi et al., 2021).

3.2.2. *Building corner extraction*

In a manner akin to the extraction of center points, corners can also be detected using a heatmap-based approach. In our Line2Poly method, we incorporate a corner prediction branch, as shown in Fig. 4, which produces a heatmap highlighting peak values corresponding to the building corners. This branch shares the same network architecture as the center point prediction branch. It is important to emphasize that the number of corner pixels is notably smaller in comparison to the rest of the image. Consequently, we utilize the focal loss function to calculate the corner points localization loss L_{cor}, which is analogous to the loss function used for center points.

3.2.3. *Filtering out redundant lines*

Within each OBB, corner points are associated with a specific building. However, inaccuracies in the prediction of bounding boxes can lead to potential loss of corner points. To mitigate this, we expand each box by a factor of 1.2. For a building with n corner points, the total number of lines formed by any two points is $(n^2\text{-}n)/2$. As a result, redundant lines need to be filtered out, a task treated as a binary classification problem where lines that do not align with ground truth lines are distinguished from the valid ones. To achieve this, we have designed a line filtering branch.

First, a line feature pooling operation is utilized to extract features for each potential line. This includes features of the two endpoints and midpoint of the potential line, as illustrated in Fig. 3. These features are then concatenated with the coordinates of the two endpoints, forming a feature representation of the line l_i, expressed as $f_i = Concat(f_{start}, f_{mid}, f_{end}, x_{start}, y_{start}, x_{end}, y_{end})$. Specifically, the line filtering branch comprises six consecutive fully connected layers with ReLU activations. The Sigmoid function is applied to generate confidence scores for each potential line, while lines exceeding a predefined threshold T_f (e.g., $T_f = 0.2$) considered valid. Finally, we obtain the preliminary feature lines of the buildings. It should be noted that the binary classification labels are assigned during the dynamic training process using the Hungarian algorithm (Stewart et al., 2016; Carion et al., 2020). This process pairs each ground truth line with the optically predicted potential line by minimizing the matching distance. The focal loss is employed to compute the line filtering loss L_f, which is similar to the center point loss.

The PLG module is overall quite lightweight, comprising only prediction heads related to bounding boxes and corners, along with a simple

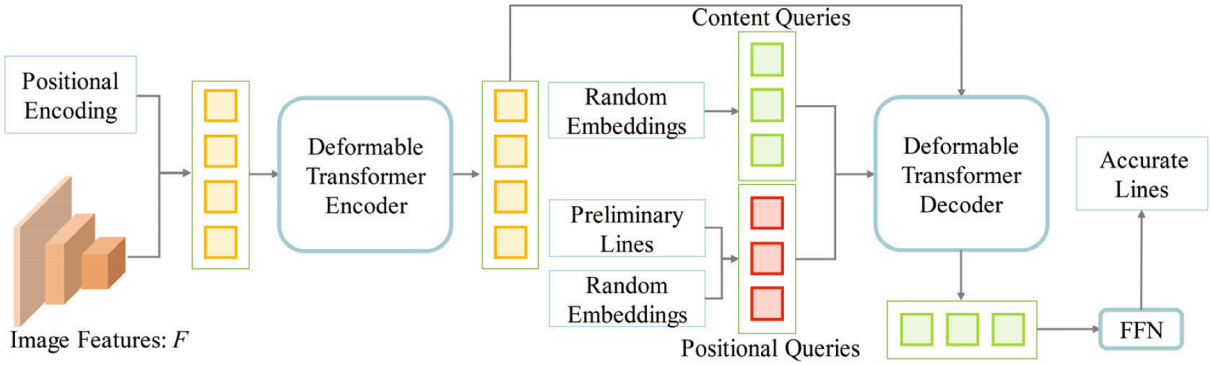

Fig. 6. The accurate line extraction (ALE) module.

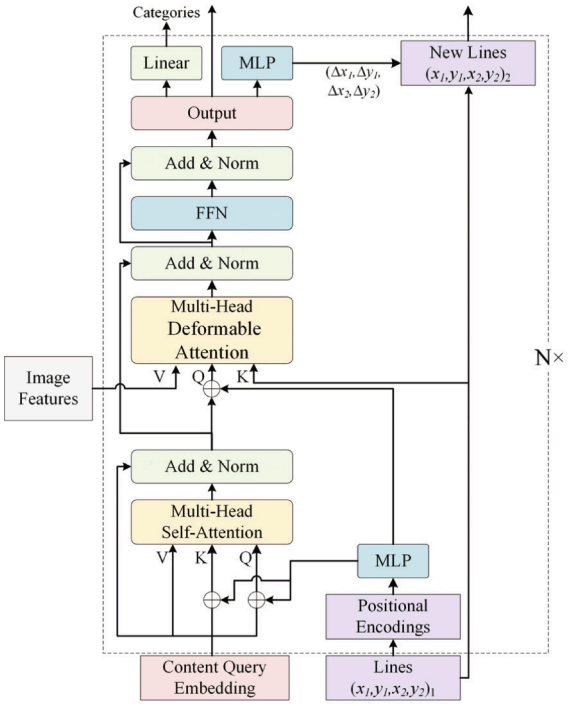

Fig. 7. Transformer decoder in the ALE module.

line filtering branch. However, it provides subsequent modules with abundant and effective positional prior information about lines.

3.3. Accurate line extraction module

Preliminary lines alone are insufficient for generating precise polygonal building footprints. To address this limitation, Line2Poly incorporates the transformer-based Accurate Lines Extraction (ALE) module, which is adapted from the object detection network DAB-DETR (Liu, et al., 2201). The ALE module leverages the preliminary lines as prior knowledge to produce highly precise building feature lines.

3.3.1. Preliminary

Before introducing our transformer-based line extraction module, we will provide a brief review of recent transformer-based object detection methods and their relations to line extraction. DETR (Carion et al., 2020) is the first model to apply transformers to object detection tasks. DETR employs a standard transformer encoder-decoder architecture, mapping deep image features extracted from a CNN backbone into a set of bounding boxes. Unlike CNN-based object detection methods, which rely on anchor boxes for feature pooling, DETR automatically extracts feature information for specific regions using learnable queries. However, the learnable query variables in DETR lack positional priors and effective supervision, leading to a slower network convergence. DAB-DETR (Liu, et al., 2201) reintroduces the anchor frame into DETR, provides location prior knowledge for the model, and guides the network to focus on specific spatial regions, which can effectively improve the feature extraction ability and convergence speed of the network.

A noteworthy parallel exists between the representations of lines and bounding boxes. As shown in Fig. 5(a), for object detection purposes, a bounding box can be represented as (x, y, w, h), where x and y denote the coordinates of the center point, while w and h represent the width and height of the bounding box, respectively. Similarly, a line segment l can be parameterized as (x_1, y_1, x_2, y_2), with (x_1, y_1) and (x_2, y_2) denoting the endpoints (Fig. 5 (b)). The objectives of line extraction and object detection are fundamentally aligned. Transformers excel in encoding long-range information, making them particularly well-suited for the extraction of elongated shapes, such as lines.

3.3.2. Accurate line extraction based on enhanced DAB-DETR

The ALE module within Line2Poly is built upon the Deformable DAB-DETR (Liu, et al., 2201), originally designed for object detection but adapted for precise feature line extraction, as shown in Fig. 6. ALE takes the multi-scale features F, derived from the CNN backbone, as its input and employs a transformer encoder for refine these features. It subsequently employs dual queries, comprising positional queries (lines) and content queries, which are fed into the decoder to iteratively extract accurate feature lines. In contrast to the original DAB-DETR, which utilizes randomly initialized embedding parameters as positional queries, ALE initializes its positional queries based on the preliminary feature lines obtained from the PLG module. This approach reduces the network's learning burden and enhances line extraction performance.

The number of preliminary feature lines may vary across images, but the number of queries should exceed the line count to ensure a satisfactory recall rate. We establish the query count as a constant k. If the line count exceeds k, only the top-k lines with the highest confidence scores are retained. Otherwise, the vacancy is compensated by introducing additional randomly-initialized positional queries. The transformation of preliminary feature lines l into positional queries Q_p is accomplished through Eq. (1), where the positional queries Q_p are derived from preliminary line l via positional encoding (PE) that utilizes sinusoidal embedding parameters generated according to the line coordinates (Liu, et al., 2201), and a multi-layer perceptron (MLP).

$$Q_p = \mathrm{MLP}(\mathrm{PE}(l)) \tag{1}$$

ALE decoder consists of six stacked layers, as shown in Fig. 7. The content query embedding and positional queries (lines) are updated layer by layer to approach the ground truth building lines. Each layer consists of both self-attention module and deformable attention module.

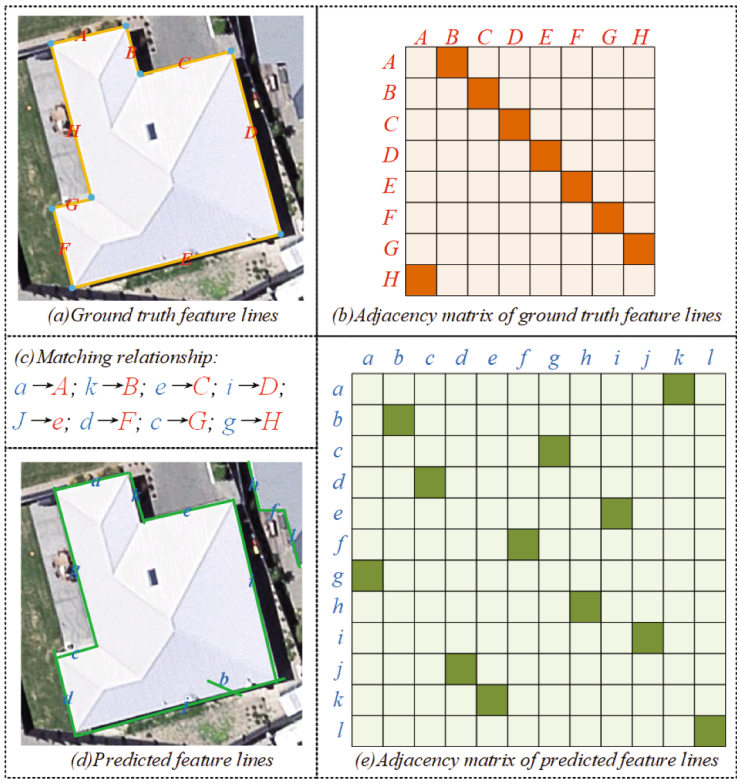

Fig. 8. The matching relationship between the ground truth and predicted feature lines of the building, along with their corresponding adjacency matrix.

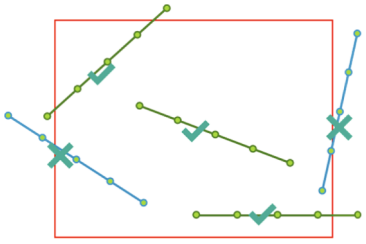

Fig. 9. Example of associating feature lines (green lines are positive and blue ones negative) to a building instance (red box). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Each module receives queries, keys, and values to conduct attention-based feature aggregation. In the self-attention, query, key and value all take content query embedding as input, while query and key contain additional positional query embedding Q_p. The output of the self-attention module is added to the input content query embedding through skip connections, processed by a layer normalization, and then used as part of the deformable attention module (Zhu et al., 2010). Deformable attention, on the other hand, employs image features as value elements, combines position encodings and content embedding that updated by the self-attention module for query elements, and aggregates neighboring feature information around the reference line for key elements. Unlike cross-attention, deformable attention focuses on pivotal feature details near the reference position, regardless of the spatial size of the feature map. This approach accelerates convergence and reduces computational overhead by efficiently leveraging sparse spatial information.

The output of each decoder layer is fed into a linear projection and a multi-layer perceptron (MLP) that composed of three stacked linear projections and ReLU operations to obtain the feature line coordinate offsets (Δx_1, Δy_1, Δx_2, Δy_2) and associated categories respectively. These offsets iteratively update the line coordinates, while the categories determine the validity of each feature line. Similar to Section 3.2.3, where potential lines are matched with ground truth, we employ the Hungarian algorithm to dynamically optimize the bipartite graph matching between predicted and ground truth lines before computing the losses during the training stage. The losses of within the ALE module include two components: the line classification loss and the line regression loss. We employ the focal loss to compute the feature classification loss L_{cls}, and the Smooth L_1 loss to calculate the line regression loss L_{lin}.

3.4. From lines to polygons

Within a building instance, the challenge of converting contour lines into polygons revolves around ascertaining the existence of adjacency relationships between any pair of lines. This can be framed as the prediction of an adjacency matrix among building lines in a clockwise manner. As shown in Fig. 8(b), feature line A is adjacent to feature line B, with the reverse adjacency relationship between them being non-existent.

3.4.1. Prediction of line adjacency matrix

Initially, we utilize the bounding box to determine the corresponding lines for each building instance. When a significant portion of line l resides within a bounding box, we link l with that specific building, as shown in Fig. 9. We evenly divide the feature line into four segments. To consider a feature line as part of the current building instance, both vertices at the 1/4 and 3/4 positions of the line must fall within the bounding box. The number of feature lines typically varies among different buildings. During training, the ground truth bounding boxes of buildings are utilized as inputs, whereas during testing, predicted bounding boxes are employed. To maintain the consistency of inputs, the number of lines for a building n_{ins} (e.g., $n_{ins} = 40$) is fixed. The rule is similar to the setting of query count k in section 3.3.2. If the line count

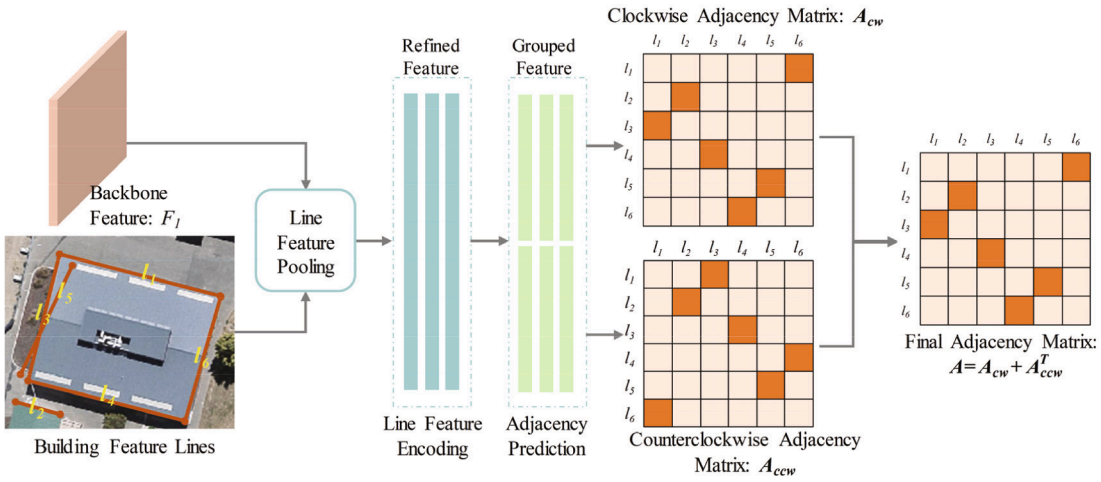

Fig. 10. Pipeline for the PTR module in Line2Poly.

Table 1
Quantitative results of feature line extraction on the VWB dataset.

method	SAP (%)	SAP_5 (%)	SAP_{10} (%)	SAP_{15} (%)
F-CLIP	57.5	47.8	60.1	64.7
L-CNN	62.1	58.1	63.5	64.8
HAWP	70.1	67.3	71.0	72.1
LETR	46.2	33.4	49.0	56.2
Line2Poly	72.3	69.6	73.1	74.1

Table 2
Quantitative results of feature line extraction on the WHU aerial building dataset.

Method	SAP (%)	SAP_5 (%)	SAP_{10} (%)	SAP_{15} (%)
F-CLIP	74.8	72.0	75.4	77.1
L-CNN	66.5	64.4	66.9	68.3
HAWP	68.7	66.6	69.1	70.4
LETR	66.8	60.5	68.3	71.6
Line2Poly	77.2	75.2	77.7	78.7

exceeds n_{ins}, only the top-n_{ins} lines with the highest confidence scores are retained. Otherwise, the vacancy is compensated by introducing additional virtual lines $l_v = (0, 0, 0, 0)$.

The bounding boxes along sometimes cannot link a building instance with all of its line elements accurately. In the refinement step, we employ the Polygon Topology Reconstruction (PTR) module to derive the adjacency matrix among lines within a given building. Leveraging this adjacency matrix, we can exclude lines that are not pertinent to the building and connect the rest lines sequentially to form the building polygon. The PTR module comprises three components, as depicted in Fig. 10: line feature pooling, line feature encoding, and adjacency matrix prediction head. In the first step, the line feature pooling operation, akin to the PLG module, is applied to extract pertinent feature information for each line. Subsequently, the line feature encoding structure refines this information through a series of three sequential layers of 1D convolution, each with a kernel size of 1, followed by ReLU activation and 1D batch normalization operations. By pairing and concatenating the refined features of n_{ins} lines, two at a time, in series, we obtain the grouped feature, denoted as $F_{group} \in \mathbb{R}^{n_{ins} \times n_{ins} \times C}$. This grouped feature is then used as input for the adjacency matrix prediction head. The adjacency matrix prediction head has two branches: one for predicting the clockwise adjacency matrix A_{cw} and the other for the counterclockwise adjacency matrix A_{ccw}. Each branch comprises three stacked layers of 2D convolution with a kernel size of 1×1, each followed by ReLU activation. The final adjacency matrix A is obtained by applying a sigmoid activation to the sum of A_{cw} and the transpose of A_{ccw}. During the training stage, we utilize the Hungarian algorithm to dynamically establish the optimal correspondences between the predicted and ground truth lines, using the distance between them as the matching cost, as shown in Fig. 8(c). Predicted lines that are not matched to any ground truth lines are only adjacent to themselves and detached from the current building. As seen in Fig. 8(e), the lines erroneously assigned to the current building (e.g., lines *f*, *h*, *l*) or incorrectly extracted (e.g., line *b*) are only adjacent to themselves and are then excluded. The PTR module utilize the focal loss to calculate the adjacency matrix prediction loss L_{adj}, and restrict supervision to valid line adjacency combinations (virtual lines l_v are not involved in the loss computation).

Finally, in the inference stage, the Sinkhorn algorithm (Cuturi, 2013; Sinkhorn and Knopp, 1967) is utilized to determine the optimal adjacency relationships among a building's contour lines based on the predicted adjacency matrix. The Sinkhorn algorithm guarantees a line adjacent to only one line. Lines that are identified as being adjacent to themselves are excluded. We proceed to calculate intersection points between adjacent lines in a clockwise manner, facilitating the transformation of the contour lines into polygons.

Table 3
Quantitative results of feature line extraction on the WHU-Mix dataset.

Method	WHU-Mix test set I (%)				WHU-Mix test set II (%)				Test set I & II (%)
	SAP	SAP_5	SAP_{10}	SAP_{15}	SAP	SAP_5	SAP_{10}	SAP_{15}	mSAP
F-CLIP	31.2	26.5	32.2	35.0	25.0	21.5	25.6	27.9	28.1
L-CNN	32.4	27.6	33.5	36.0	19.4	16.6	19.8	21.8	25.9
HAWP	37.1	31.6	38.4	41.2	22.5	19.5	23.0	24.9	29.8
LETR	28.2	17.8	30.2	36.6	20.0	13.2	21.1	25.7	24.1
Line2Poly	36.6	32.1	37.8	40.0	25.8	22.6	26.4	28.3	31.2

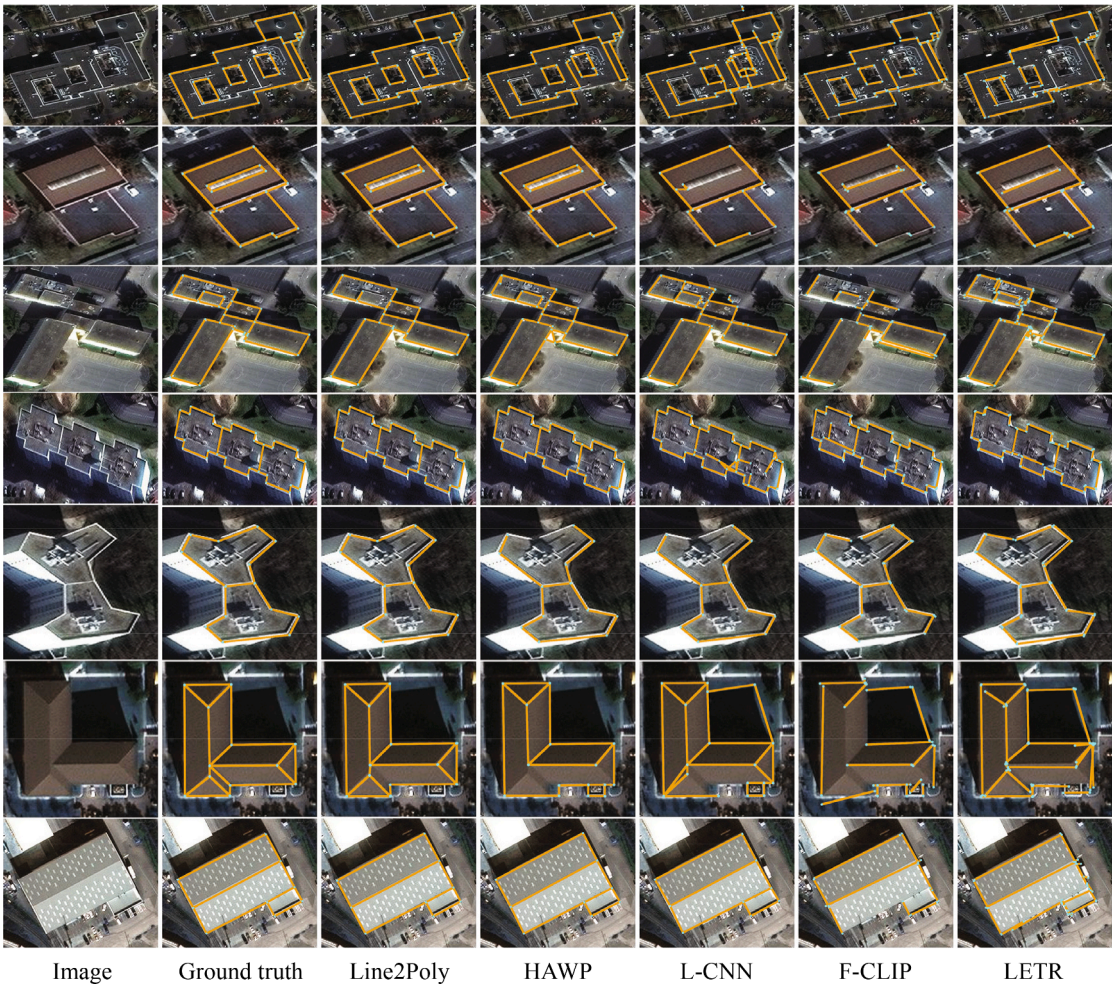

Fig. 11. Qualitative comparison on VWB dataset among different feature line extraction methods.

Table 4
Quantitative results of building extraction on the WHU aerial building dataset.

Method	AP (%)	AP_{50} (%)	AP_{75} (%)	AR (%)
Mask R-CNN	65.3	90.0	77.1	70.7
Yolact	65.3	88.5	76.5	71.1
Solo	68.6	89.8	79.4	73.4
Deep Snake	72.7	91.5	82.8	78.6
CLP-CNN	72.6	90.9	82.6	78.0
BuildMapper	73.6	89.0	81.6	78.9
Line2Poly	73.8	89.4	82.1	79.7

4. Experimental settings

4.1. Datasets

To demonstrate the superiority and robustness of Line2Poly for building polygon extraction tasks, we choose three open source building datasets with varying styles, the Vectorizing World Buildings (VWB) dataset (Nauata and Furukawa, 2020), the WHU aerial building dataset (Ji et al., 2018), and the WHU-Mix (vector) building dataset (Wei et al., 2023), for experiments.

4.1.1. Vectorizing World buildings dataset

The VWB (Nauata and Furukawa, 2020) dataset provides intricate data regarding roof structural feature lines for individual buildings. This dataset encompasses 2,001 256 × 256-pixel aerial image tiles of individual buildings, 1,010 tiles in Atlanta, 670 tiles in Paris, and 321 tiles in Las Vegas. The training set consists of 1,601 tiles, while the test set comprises 400 tiles.

4.1.2. WHU aerial building dataset

The WHU aerial building dataset (Ji et al., 2018) offers a large quantity of high-resolution aerial imagery with accurate annotations of building polygons. Covering a vast area of 200 km^2 in Christchurch, New Zealand, the dataset comprises over 187,000 buildings representing diverse architectural styles and purposes, all captured at a resolution of 0.2 m. The aerial images have been cropped into 5,640 512 × 512 tiles. The dataset is categorized into three subsets: a training set consisting of 2,793 tiles (about 130,000 buildings), a validation set comprising 627 tiles (about 14,500 buildings), and a test set containing 2,220 tiles (about 42,000 buildings).

4.1.3. Whu-mix (vector) building dataset

The WHU-Mix (vector) building dataset (Wei et al., 2023; Luo et al., 2208), hereinafter referred to as the WHU-Mix dataset, comprises remote sensing imagery of buildings in multiple countries across five continents, showcasing significant variations in building styles and types. This dataset facilitates comprehensive and effective performance evaluation of various methods. It consists of over 64,000 tile images, representing more than 754,000 individual buildings and covering a total geographical area of approximately 1,100 km^2. The WHU-Mix dataset has been divided into four distinct subsets: a training set containing 43,778 tiles, a validation set with 2,922 tiles, test set I containing 11,675 tiles, and test set II comprising 6,011 tiles. It should be noted that

Fig. 12. Examples of feature line and polygonal building results achieved by Line2Poly on the WHU aerial building dataset. The left side of each group of images shows the feature line extraction result, and the right side displays the building polygon reconstruction result.

Table 5
Quantitative results of building extraction on the WHU-mix dataset.

Method	WHU-Mix test set I (%)				WHU-Mix test set II (%)				Test set I & II (%)	
	AP	AP_{50}	AP_{75}	AR	AP	AP_{50}	AP_{75}	AR	mAP	mAR
Mask R-CNN	47.0	67.0	53.2	53.7	46.1	73.9	49.0	54.8	46.5	54.3
Yolact	42.3	65.7	47.2	49.7	41.3	71.3	42.3	50.6	41.8	50.2
Solo	57.1	83.2	65.1	63.9	45.3	74.3	47.9	54.7	51.2	59.3
Deep Snake	55.3	82.1	63.0	61.8	46.9	73.9	51.5	54.8	51.1	58.3
CLP-CNN	55.6	81.8	63.5	62.3	48.3	75.2	52.4	56.4	52.0	59.4
BuildMapper	59.1	83.8	67.3	65.6	48.8	73.0	53.2	56.6	54.0	61.1
Line2Poly	58.6	81.9	66.2	65.5	48.9	73.3	52.8	58.4	53.8	62.0

test set II exhibits no geographical overlap with the training set, thereby introducing a heightened level of difficulty and facilitating a more comprehensive assessment of the method's generalization capacity.

4.2. Evaluation metrics

We conducted a comprehensive comparison of different methods using multiple evaluation metrics to assess their performance from different perspectives. Specifically, three key aspects, feature line extraction, polygonal building extraction, and manual-level building polygon extraction, are evaluated.

4.2.1. Feature line extraction evaluation metric

We utilize Structural Average Precision (SAP) (Zhou et al., 2019) as the evaluation metric to assess the performance of feature line extraction. SAP is defined as the area under the precision-recall curve, which is computed based on a scored list of the detected feature lines. Feature line $l=(p_1, p_2)$ is considered correctly detected if it satisfies the conditions in Eq. (2), where $\widehat{l}=(\widehat{p}_1,\widehat{p}_2)$ represents the corresponding ground truth feature line, $(i, j)=(1, 2)$ or $(2, 1)$, and ϑ is a predefined threshold. To prevent duplicate predictions, each ground truth feature line is associated with only one prediction, with repeated predictions being deemed errors. Consistent with previous studies on line extraction (Zhou et al., 2019; Xue, 2020; Dai et al., 2022), we calculate the SAP at a resolution of 128 × 128 pixels, and set the threshold ϑ to 5, 10, and 15, respectively, resulting in the accuracy of SAP_5, SAP_{10}, and SAP_{15}. The mean SAP (mSAP) represents the average accuracy across the three threshold conditions.

$$\min_{(i,j)} \|p_1 - \widehat{p}_i\|^2 + \|p_1 - \widehat{p}_j\|^2 \le \vartheta \tag{2}$$

4.2.2. Instance-level building extraction evaluation metric

The evaluation of instance-level building extraction quality employs the standard COCO measure, which includes average precision (AP) and average recall (AR) calculated at various intersection over union (IoU) thresholds. The IoU value is determined by the ratio of the intersection and the union of the predicted and ground-truth building area. AP is computed as the average precision across 10 IoU values ranging from 0.50 to 0.95 with a step size of 0.05. AR is calculated in a similar manner to AP. Additionally, we report AP_{50} and AP_{75}, representing the average precision at IoU thresholds of 0.5 and 0.75, respectively.

4.2.3. Manual-level building polygon extraction evaluation metric

Follow (Wei et al., 2019; Wei et al., 2023), the evaluation metric for building extraction at the manual level is determined by the Valid Polygon Ratio (VPR). The VPR represents the ratio of automatically extracted building vector polygons that align with the level of manual delineation to the total number of ground truth building polygons. Specifically, the level of manual delineation is defined as the accuracy of boundary delineation that a human operator can reasonably achieve, with an approximate bias of 2- or 3-pixels bias (as it is challenging for a human to guarantee delineation at a one-pixel or subpixel level). This tolerance value can be used to calculate an IoU value between the "tolerance" polygon and ground truth polygon. The IoU value serves as the threshold to determining the quality of manual-level building polygon extraction.

Fig. 13. Examples of feature lines and polygons achieved by Line2Poly on the WHU-Mix dataset. The left side of each group of images represents the feature line extraction result, and the right side displays the building polygon reconstruction result.

4.3. Implementation details

The proposed Line2Poly method optimizes several task losses collectively, encompassing the bounding box loss L_{obb}, corner loss L_{cor}, line filtering loss L_f, feature line regression loss L_{lin}, feature classification loss L_{cls}, and adjacency matrix loss L_{adj}. Consequently, we define the multi-task loss of Line2Poly method as:

$$L = L_{obb} + L_{cor} + L_f + \lambda L_{lin} + L_{cls} + L_{adj} \tag{3}$$

where λ is the balancing hyperparameter for L_{lin}, and we set λ to 5. We implemented and tested Line2Poly method in PyTorch on a desktop computer with a Nvidia RTX 3090 24 GB graphics processing unit (GPU). During the training stage of the Line2Poly method, we employ data augmentation techniques including multi-scale scaling, flipping, cropping, and color dithering. We utilize the Adam function with a learning rate of 1e-4 as optimizer to optimize the entire network.

5. Results and discussion

5.1. Comparison with the state-of-the-art feature line extraction methods

Accurate extraction of building feature lines is a fundamental prerequisite for building polygon reconstruction in this study. In this section, we conduct a comparative analysis between Line2Poly and state-of-the-art methods for feature line extraction. Tables 1-3 present a quantitative comparison of Line2Poly alongside F-CLIP (Dai et al., 2022), L-CNN (Zhou et al., 2019), HAWP (Xue, 2020), and LETR (Xu et al., 2021), focusing on SAP, SAP_5, SAP_{10}, SAP_{15} metrics, across the VWB dataset, WHU aerial building dataset, and WHU-Mix dataset.

As depicted in Table 1, Line2Poly achieves the highest accuracy, with a 72.3 % SAP score on the VWB dataset, surpassing the second-ranking HAWP method by 2.2, and outperforming other feature line extraction methods by more than 10 %. It's noteworthy that F-CLIP, L-CNN, and HAWP are all CNN-based methods, which are constrained by the limited capacity of CNN architecture to encode long-range information. Specifically, HAWP employs a reparameterization strategy for feature lines, establishing connections between lines and regions to enhance network learning, resulting in the second-best accuracy. LETR, on the other hand, is a fully transformer-based method designed based on DETR. However,

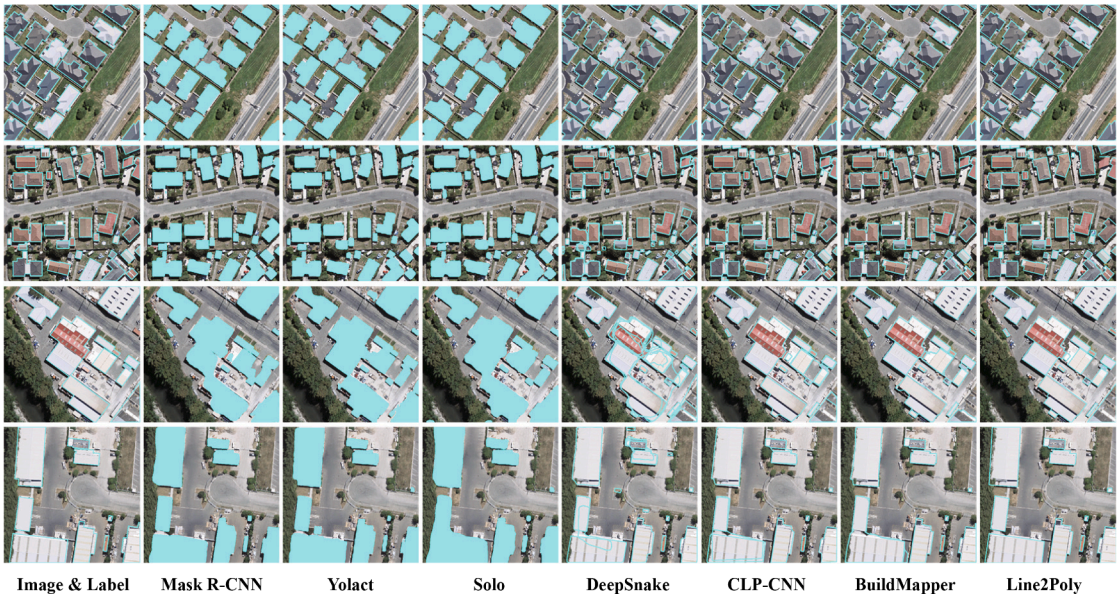

Fig. 14. Qualitative comparison on the WHU aerial building dataset.

its performance is hampered by the limited training data available from the VWB dataset, which comprises only 1601 256 × 256-pixel tiled images for training. Consequently, LETR exhibits the lowest overall accuracy among all evaluated methods. Line2Poly's two-stage feature line extraction strategy combines the strengths of both CNN and Transformer techniques. The prior information generated by the CNN-based PLG module mitigates the training challenges of the transformer-based ALE module. In turn, the ALE module compensates for the limitations of the CNN-based module in encoding long-range information. This synergistic approach propels Line2Poly to achieving the highest level of precision performance.

Fig. 11 presents examples of feature line extraction results from different methods on the VWB dataset. Line2Poly stands out with its excellent performance, closely matching the ground truth, and exhibiting minimal missing or incorrect detections. HAWP secures the second position but occasionally suffers issues of missing lines. F-CLIP and L-CNN demonstrate similar and moderate performance. In contrast, LETR lags behind, displaying relatively low accuracy in feature line positioning. As an example, in the sixth row of Fig. 11, F-CLIP, LCNN, and LETR misinterpret the shadow area as the building edge, while Line2Poly and HAWP correctly distinguish this situation. However, HAWP misses five ridge lines and Line2Poly misses one in the left-lower low-contrast region. This illustrates the common challenge of extracting lines in low-contrast scenarios.

To further affirm Line2Poly's superiority in feature line extraction, we conducted additional experiments on the WHU aerial building dataset and the WHU-Mix dataset. These datasets solely provide building contour polygon information, so we transformed them into lines for feature line extraction. As shown in Table 2, on the WHU aerial building dataset with higher data quality, Line2Poly maintains its top-notch accuracy performance. L-CNN, HAWP, and LETR deliver very similar overall accuracy performances. Unlike the VWB dataset, the WHU aerial building dataset offers a larger number of training samples, enabling LETR to be fully trained and achieve performance on par with CNN-based methods in terms of accuracy. Notably, F-CLIP secures the second-best accuracy, closely trailing Line2Poly.

The WHU-Mix dataset combines data from various geographical regions and different sensors, demanding more robust algorithm capabilities. Moreover, the WHU-Mix test set II is entirely distinct from the training set, ensuring no geographical adjacency or overlap, facilitating a robust evaluation of methods' generalization ability on heterogeneous remote sensing images. As shown in Table 3, Line2Poly demonstrates the best overall accuracy performance (mSAP) on the WHU-Mix dataset. Although HAWP outperforms Line2Poly by 0.5 % SAP on the test set I, it falls behind Line2Poly by 3.3 % SAP on the test set II. This discrepancy highlights HAWP's proficiency with homologous data but its limitations in generalization for complex building feature line extraction tasks in practical applications. In contrast, F-CLIP exhibits robust generalization capabilities due to its network's resistance to overfitting. On the other hand, L-CNN and LETR perform less satisfactorily results overall.

5.2. Comparison with instance extraction methods

We conduct a comparative evaluation of Line2Poly alongside state-of-the-art building instance extraction methods on both the WHU aerial building dataset and the WHU-Mix dataset. These methods include Mask R-CNN (He et al., 2017); Yolact (Bolya et al., 2019), Solo (Wang et al., 2020), Deep Snake (Peng et al., 2020), CLP-CNN (Wei et al., 2021), and BuildMapper (Wei et al., 2023).

Table 4 provides a quantitative comparison of building extraction performance on the WHU aerial building dataset. CLP-CNN, BuildMapper, and Line2Poly are methods purpose-built for building extraction, equipped to generate regular building polygons. Line2Poly stands out with the highest accuracy, achieving 73.8 % AP and 79.7 % AR. BuildMapper secures the second position and closely rivals Line2Poly. In contrast, common instance segmentation methods such as Mask R-CNN, Deep Snake, Yolact, and Solo lack specialized approaches for the inherent regularity of building structures, leading to comparatively inferior results. Deep Snake achieves a 72.7 % AP, whereas Mask R-CNN, Yolact and Solo exhibit lower performance, with AP values of 65.3 %, 65.3 % and 68.6 %, respectively. It's worth noting that Deep Snake, CLP-CNN, and BuildMapper are all contour-based methods capable of directly generating polygonal buildings. However, contour-based methods require additional modules or post-processing steps to eliminate redundant vertices based on empirical thresholds. Moreover, we observed limitations in these techniques when dealing with complex building scenarios. In contrast, Line2Poly utilizes feature lines for polygon generation, effectively avoiding vertex redundancy and directly yielding regularized building polygons.

Fig. 12 provides visual examples of feature line and polygon extraction results achieved by Line2Poly on the WHU aerial building dataset. Within the high-quality WHU aerial building dataset, Line2Poly

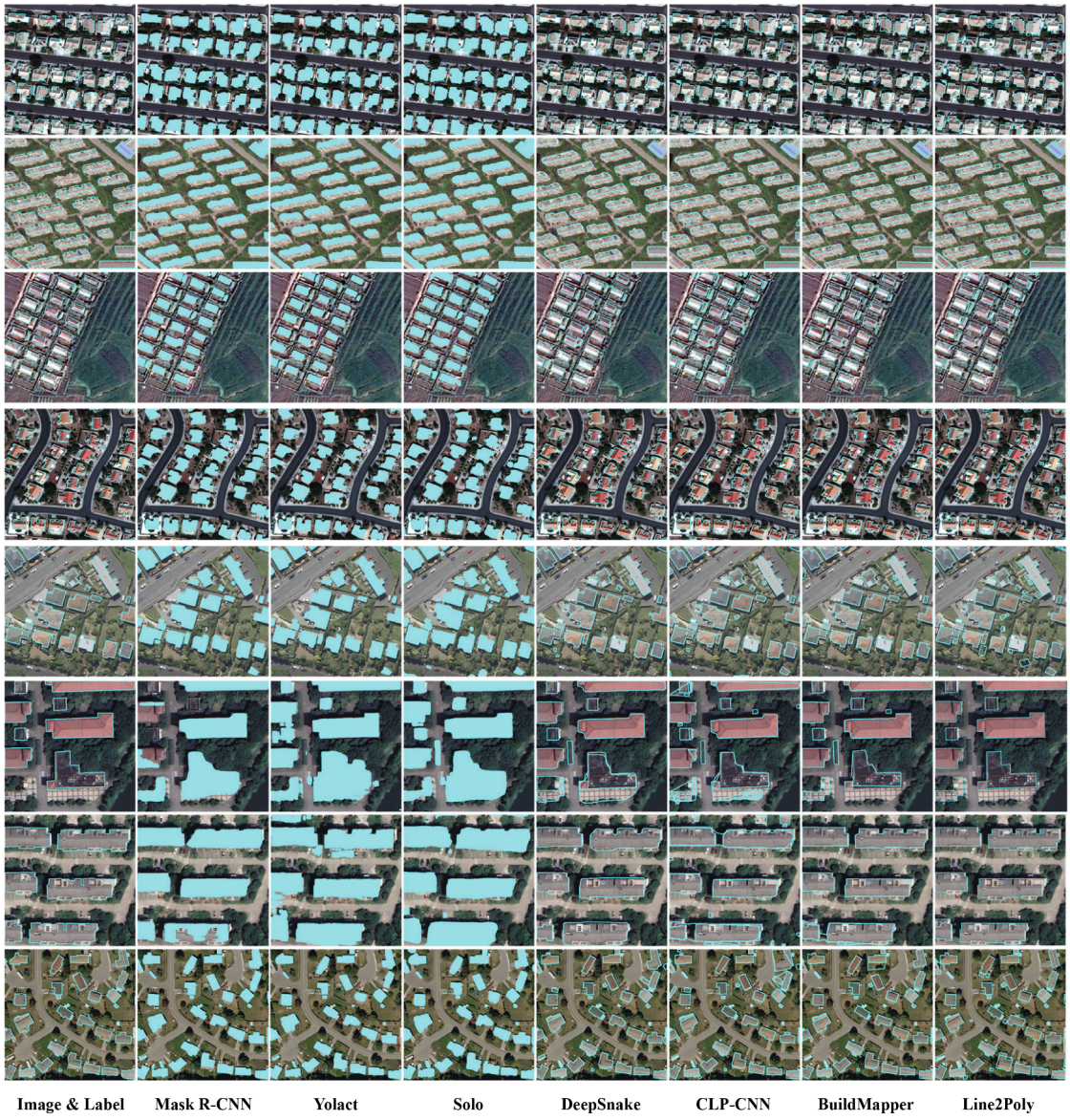

Fig. 15. Qualitative comparison on the WHU-Mix dataset.

Table 6
The percentage of extracted polygonal buildings that reach the manual delineation level with different methods.

Method	WHU		WHU-Mix test set I		WHU-Mix test set II	
	2-Pixel	3-Pixel	2-Pixel	3-Pixel	2-Pixel	3-Pixel
MA-FCN	68.2 %	77.5 %	51.4 %	63.6 %	29.5 %	40.1 %
CLP-CNN	69.3 %	85.5 %	57.1 %	73.7 %	49.7 %	65.5 %
BuildMapper	82.2 %	87.1 %	66.1 %	77.2 %	53.4 %	62.9 %
Line2Poly	83.2 %	87.4 %	68.7 %	79.5 %	59.7 %	71.5 %

inherently ensures result regularity, closely resembling manual delineation. Furthermore, Line2Poly's learnable polygon topology reconstruction (PTR) module effectively identifies inaccurate or redundant feature lines as invalid and construct complete polygons, as illustrated within the red box in the upper-left image of Fig. 12.

Table 5 presents the building extraction accuracy across various methods on the WHU-Mix dataset, with mAP (mean average precision) and mAR (mean average recall) for both test set I and test set II. Line2Poly's outstanding performance on the WHU-Mix dataset is evident. A comprehensive comparison between result from test set I and test set II highlights Line2Poly and BuildMapper as the leading contenders among the methods evaluated. They exhibit close mAP (53.8 vs. 54.0), with Line2Poly surpassing BuildMapper by 0.9 % in mAR. In test set I, Line2Poly obtains the same AR as BuildMapper and falls marginally behind by 0.5 AP. However, in the more practical and challenging test set II, Line2Poly achieves the same AP as BuildMapper but outperforms it by 1.8 AR. A detailed comparative analysis between Line2Poly and BuildMapper will be conducted in subsection 5.5.1.

Fig. 13 displays Line2Poly's results on the WHU-Mix dataset, with the upper two rows derived from test set I and the lower two rows from test set II. Line2Poly consistently delivers satisfactory results across diverse building types. However, due to the variable image quality in the WHU-Mix dataset, Line2Poly may encounter occasional false extractions, omissions, and imprecise localizations of feature lines. Despite these challenges, Line2Poly's robust topology reconstruction module consistently achieves accurate building polygon extraction.

Fig. 14 and Fig. 15 show examples of results from different methods on the WHU and WHU-Mix dataset, respectively. Overall, common instance segmentation methods such as Mask R-CNN, Yolact, and Solo yield coarse raster building maps, they cannot process the tree occlusion problem, as shown in the third row of Fig. 14. DeepSnake and CLP-CNN

Fig. 16. Examples of building polygons achieved by Line2Poly on the WHU-Mix test set II.

Table 7
The effect of the PLG and ALE modules of Line2Poly for feature line extraction.

Method	SAP (%)	SAP_5 (%)	SAP_{10} (%)	SAP_{15} (%)
Only PLG	64.1	65.2	64.2	62.9
Only ALE	74.3	72.3	74.8	75.9
PLG and ALE	77.2	75.2	77.7	78.7

can directly produce polygons but with obvious boundary errors in large buildings (see the last two rows of Fig. 14 and the sixth row of Fig. 15). BuildMapper creates much more satisfactory polygons with occasionally failed cases. The proposed Line2Poly demonstrates precise extraction of regular polygonal buildings across diverse scenarios, surpassing alternative methods.

5.3. Comparison of manual-level building extraction methods

Table 6 displays the VBP accuracy of various regular polygonal building extraction methods, including Line2Poly, MA-FCN (Wei et al., 2019), CLP-CNN (Wei et al., 2021), and BuildMapper (Wei et al., 2023), under 2 or 3-pixel-accuracy assumptions.

On the WHU dataset, MA-FCN, which comprises a semantic segmentation network and empirical regularization post-processing, achieves 68.2 % and 77.5 % of building polygons reaching a manual delineation level under 2-pixel and 3-pixel accuracy, respectively. This method necessitates additional raster-to-vector processing, resulting in precision loss and a relatively lower degree of automation. CLP-CNN incorporating a vertex-based network and post-processing regularization, reports 69.3 % at 2-pixel accuracy and 85.5 % at 3-pixel accuracy. Unlike MA-FCN and CLP-CNN, which rely on empirical regularization algorithms, the recent BuildMapper achieves end-to-end extraction of vectorized building polygons, resulting in 82.2 % at 2-pixel accuracy and 87.1 % in 3-pixel accuracy. Our proposed Line2Poly, also an end-to-end method, attains 83.2 % at 2-pixel accuracy, outperforming BuildMapper by 1.0 %, surpasses it by 0.3 % at 3-pixel accuracy. In compared to the early MA-FCN approach, Line2Poly demonstrates significant improvements of 15.0 % and 9.9 % under the 2-pixel and 3-pixel accuracy criteria, respectively. According to previous researches (Wei et al., 2023; Wei et al., 2021), a 3-pixel precision level is deemed sufficient for evaluating manual delineation quality. Therefore, the proposed Line2Poly method effectively extracts the vast majority (up to 87 %) of buildings polygons from high-quality remote sensing imagery, demonstrating potentials for practical applications like map updates.

The WHU-Mix dataset encompasses a wide variety of complex building types, which poses a challenge for all methods compared to the WHU aerial building dataset. In particular, the WHU-Mix test set II simulates real-world applications, where models are pretrained on a large dataset and then evaluated on previously unseen images from different cities. Therefore, test set II represents a formidable challenge and serves as a robust reference for assessing the practical performance of various methods. Despite these difficulties, Line2Poly exhibits exceptional performance when compared to other methods. In the WHU-Mix test set I, 68.7 % and 79.5 % of polygons achieve 2-pixel and 3-pixel delineation accuracy, respectively, marking an improvement of 2.6 % and 2.3 % over the suboptimal BuildMapper. In test set II, 59.7 % and 71.5 % of buildings reach 2-pixel and 3-pixel accuracy, surpassing the BuildMapper method by 6.3 % and 8.6 %, respectively. These results in challenging contexts are highly promising, highlighting Line2Poly's potential effectiveness in real-world applications, where over 71 % of buildings can be automatically labeled without manual intervention.

Fig. 16 further demonstrates Line2Poly's capability to extract buildings polygons with different styles and architectures from images of varying quality in the WHU-mix test set II. Whether dealing with a residential house, a large-scale factory, or a towering skyscraper, the proposed method consistently and accurately extracts their regularized polygons. It's worth noting that certain buildings with more complex shapes (e.g., the one in the bottom-right of Fig. 16) or those captured in non-orthorectified images often exhibit multiple principal directions. Regularization algorithms, which assume that building edges align vertically with the principal direction, as used in MA-FCN and CLP-CNN, struggle to generate accurate polygons under these circumstances. Additionally, CNN-based methods may face limitations in dealing with large buildings (e.g., the one in the bottom-left of Fig. 16) due to limited receptive fields. In contrast, Line2Poly adeptly handles these scenarios by reconstructing polygons from building lines.

Fig. 17. Examples of feature line extraction results based on various modules.

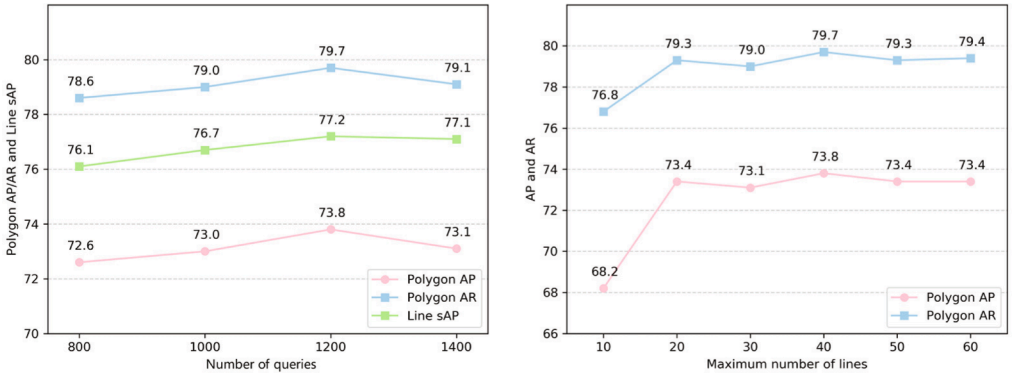

Fig. 18. Left: Impact of query numbers in the ALE module; Right: Effect of line count in the PTR module.

5.4. Ablation study

In this subsection, we further investigate the details of the proposed Line2Poly method, including the effectiveness of the feature line extraction component, the influence of query numbers within the ALE module, and the influence of line count in the PTR module. All experiments were conducted using the WHU aerial building dataset.

5.4.1. The effect of PLG and ALE module for feature line extraction

We first evaluate the contribution of the preliminary feature line generation (PLG) module and the accurate feature line extraction (ALE) module. In Table 7, "Only PLG" denotes the exclusive use of the CNN-

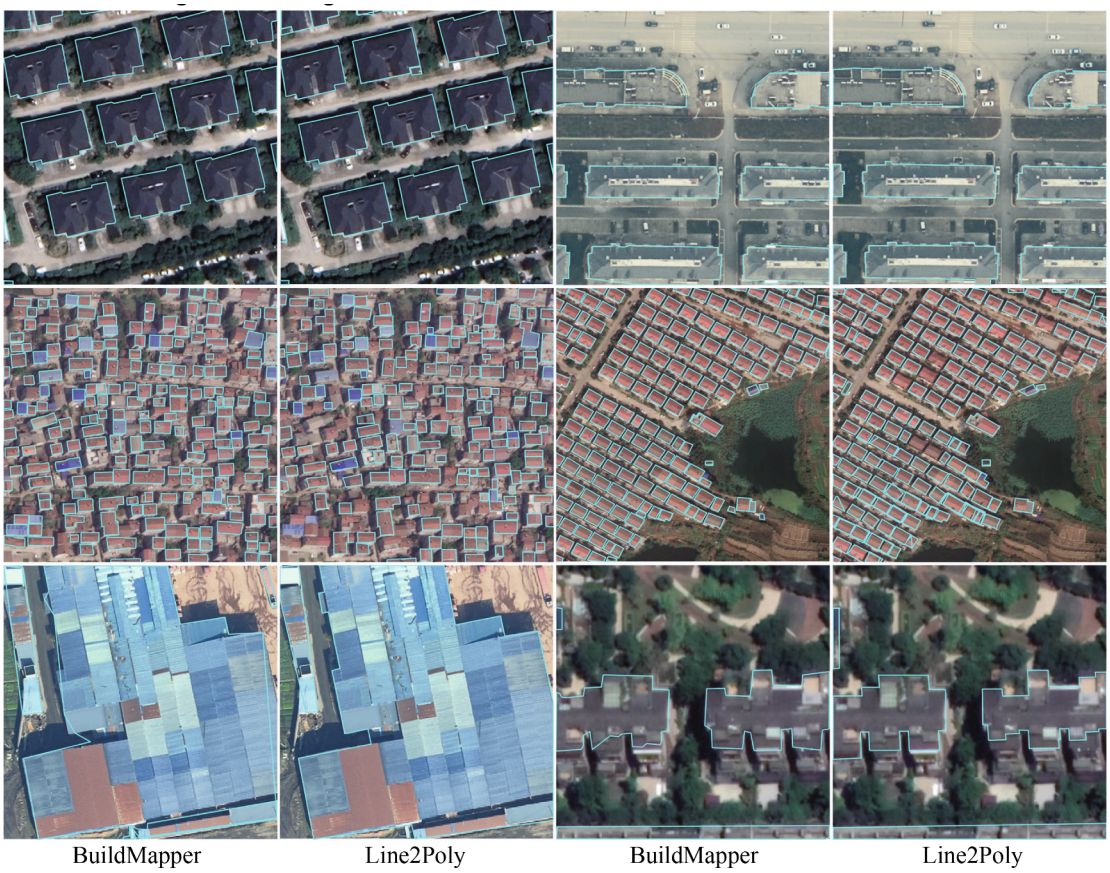

Fig. 19. Results of Line2Poly and BuildMapper in different building-style scenarios. The first row: common building areas; the second row: densely distributed building scenes; the third row: large-sized buildings zones.

based PLG module for feature line extraction, while "Only ALE" involves utilizing the transformer-based ALE module with randomly initialized embeddings as positional queries. The combined use of both PLG and ALE modules is labeled as "PLG and ALE". Using only the PLG module achieves 64.1 % SAP, comparable to the performance levels observed in CNN-based methods such as L-CNN and HAWP in Table 2. Conversely, the exclusive utilization of the ALE module results in a 74.3 % SAP, demonstrating a remarkable 10.2 % improvement over the use of the PLG module alone. This observation indicates the Transformer's superiority over CNN in the context of feature line extraction, attributed to its potent capacity for encoding global information. The Line2Poly approach, which sequentially integrates the PLG and ALE modules, achieving a 77.2 % SAP. This is accomplished by initializing positional queries within the ALE module, originally randomized, with prior feature line information obtained from the PLG module.

Fig. 17 shows examples of feature line extraction outcomes employing diverse methodologies across three distinct building scenarios: residential zones (characterized by small-sized buildings), villa districts (with mid-sized buildings), and industrial areas (featuring large-sized buildings). Notably, significant line extraction errors are evident when relying solely on the CNN-based PLG module. Conversely, feature lines generated by the transformer-based ALE module exhibit notably higher visual quality, albeit with some remaining errors. The integration of both modules obtains perfect performance in these scenarios.

5.4.2. Impact of query numbers in the ALE module

The quantity of queries k plays a crucial role in influencing the performance of Line2Poly's ALE module. As shown on the left side of Fig. 18, starting with an initial query count of 800 and gradually increasing it has demonstrated an enhancement in precision for both building feature line and polygon extraction, with the peak performance achieved at 1200 queries. However, further increments in the query count resulted in a slight decrease in accuracy. This observation highlights the fact that an excessive number of positional queries can impact the network's ability to leverage prior information from preliminary feature extraction. Consequently, we have fixed the positional query count k at 1200 for optimal results.

5.4.3. The influence of line number in the PTR module

In the case of more intricate buildings, a higher number of lines is required to accurately delineate polygons. The right side of Fig. 18 shows the relationship between polygon extraction accuracy and the feature line count n_{ins} within the PTR module. It can be observed that accuracy stabilizes as n_{ins} exceeds 20, with the peak accuracy achieved when n_{ins} equals 40. To inform this decision, we conducted a thorough analysis of line counts for individual buildings in both the WHU and the WHU-Mix dataset, revealing that 99.82 % and 99.55 % of buildings, respectively, have fewer edge lines than 40. Hence, we adopted 40 as the maximum allowable number of lines in this paper.

5.5. Discussion

5.5.1. Comparison of Line2Poly and BuildMapper

Both Line2Poly and BuildMapper are end-to-end methods tailored for polygonal building extraction. Previous experiments conducted on two datasets have consistently demonstrated the exceptional performance of both approaches in comparison to other methods. To facilitate a more comprehensive comparative analysis between Line2Poly and BuildMapper, their results are visually presented in Fig. 19, encompassing diverse architectural scenarios, including common building areas (first row), densely building scenes (second row), and large-size

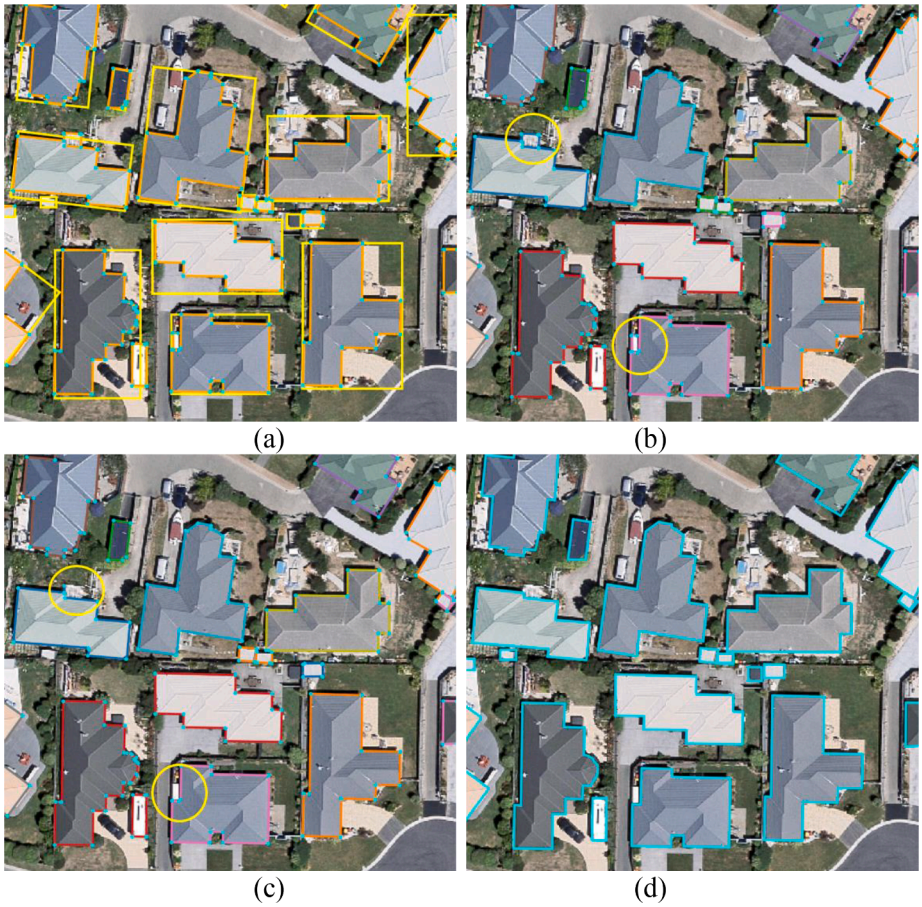

Fig. 20. From building feature lines to polygons. (a) Building feature lines and bounding boxes; (b) lines corresponding to each building instances; (c) removal of isolated lines; (d) building polygon reconstruction.

buildings zones (third row). The comparison shown in Fig. 19 demonstrates that Line2Poly excels in generating more regularized polygons, ultimately resulting in superior overall performance when contrasted with BuildMapper. In the first row of Fig. 19, both methods perform on par. In the second row of Fig. 19, the densely arranged buildings and closely positioned boundaries present challenges for the algorithms, a situation that can be intricate even for human operators. However, both Line2Poly and BuildMapper successfully and accurately delineate the majority of individual buildings. BuildMapper employs a strategy of regressing building contour coordinates from central points, thus advantageous in extracting small-sized buildings. Line2Poly is constrained by the limited number of feature lines available per image, leading to occasional missed detections. In scenes with large-sized buildings, as illustrated in the bottom row of Fig. 19, another challenge for building extraction arises. Large-sized buildings often are divided into multiple tiles of limited size. Extracting these large-sized buildings necessitates the network's ability to encode long-range information. Due to the confined receptive field of the CNN architecture, BuildMapper encounters difficulties in accurately extracting large-sized buildings. In contrast, Line2Poly, integrated with the Transformer architecture, demonstrates robust global information encoding capabilities and holds a distinct advantage in handling such scenarios.

5.5.2. *Influence of feature line extraction error on polygon reconstruction*

Fig. 20 illustrates the process of translating lines into polygons. Fig. 20 (a) shows the results of building feature line and bounding box extraction. Based on the positions of lines and boxes, we can initially link the lines to each building instance (Fig. 20 (b)), represented with distinct colors. Subsequently, the PTR module is employed to predict the adjacency relationship matrix between lines for each building. Those isolated lines (adjacent to themselves in the matrix) are then excluded, as shown in the yellow ellipses in Fig. 20 (c). Ultimately, the reconstruction of building polygons is achieved according to the valid lines and their order recorded in the adjacency relationship matrix (Fig. 20 (d)).

We can see the key step of line-to-polygon reconstruction is how to handle the extracted separate feature lines with potential errors. Fortunately, our dedicatedly designed PTR module in Line2Poly can effectively manage several types of line errors. As more examples, the top row of Fig. 21 highlights the PTR module's capability to rectify misidentified or redundant feature lines. Those erroneous lines, marked with yellow circles, are identified as "self-adjacent" and subsequently eliminated, thus reinstating the correct topological relationships among valid lines. In the second row of Fig. 21, enclosed in orange circles, are instances where certain feature lines were either missed or duplicated during detection. For those missing lines, the PTR module strategically seeks the optimal adjacent line for each line within the current set of valid lines, ensuring polygon closure. Moreover, in cases where absent lines result in the approximation of parallelism between two adjacent feature lines, the PTR module introduces a perpendicular line to guarantee polygon closure.

5.5.3. *Limitation and future work*

While Line2Poly has showcased impressive performance in polygonal building extraction, it does have some limitations. One common issue arises from occlusions caused by surrounding objects. In cases where the occluded region of a building is relatively small, Line2Poly can infer the missing parts and complete the contours (Fig. 22 (a)).

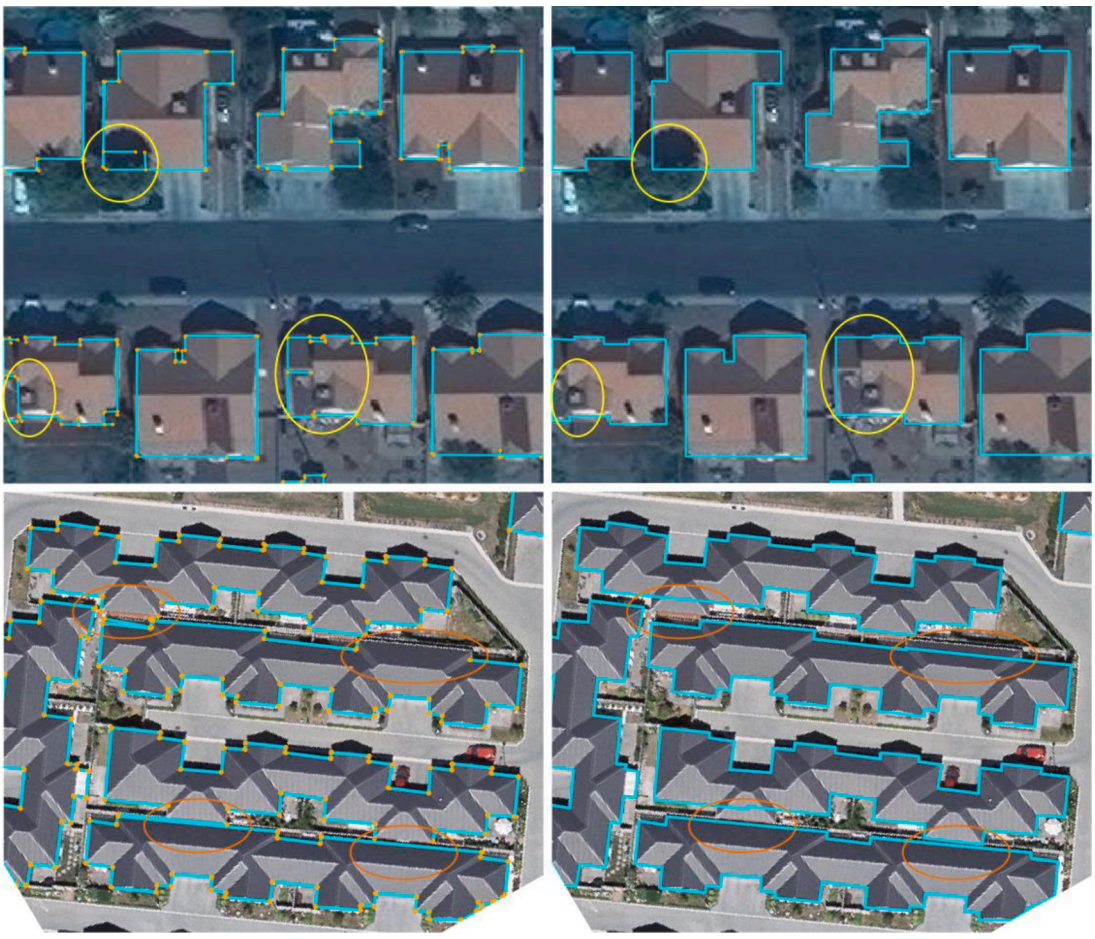

Fig. 21. Line2Poly's polygon topology reconstruction results in cases involving erroneous feature line extraction. In the upper row, false feature line extractions are highlighted by yellow circles, while in the lower row instances of missing lines are marked with orange circles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

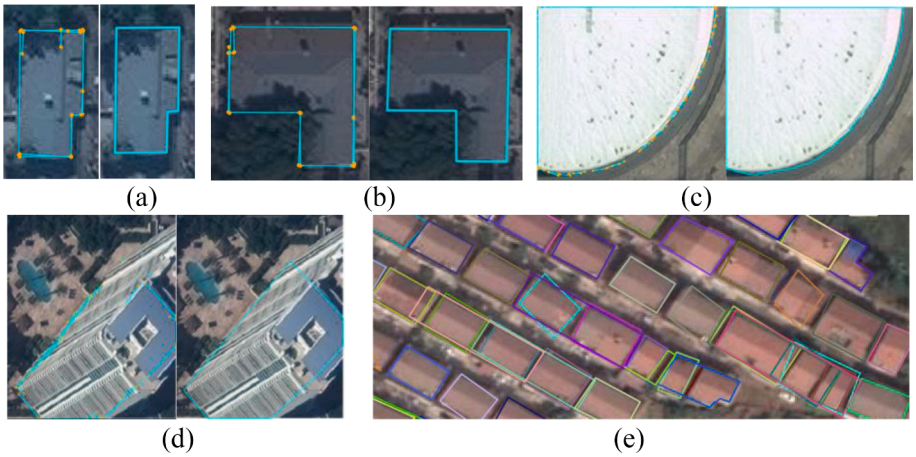

Fig. 22. Some examples of Line2Poly in special scenarios.

However, when the occluded region is extensive (Fig. 22 (b)), Line2Poly can only provide a rough shape. Additional assistance from sources such as Digital Surface Models (DSM) can mitigate this problem. Another limitation is that Line2Poly is primarily designed for polygonal buildings. For round-shaped buildings (as seen in Fig. 22 (c)), Line2Poly approximates the curve with a series of short lines. Thirdly, while Line2Poly can tolerate minor errors in feature line extraction, it cannot rectify significant errors. For example, when the predicted feature lines for a large building with an unusual tilt angle in the WHU-Mix dataset result in bad boundary (Fig. 22 (d)), Line2Poly cannot provide a precise solution. Fig. 22 (e) showcases a densely populated region with poor imaging quality, where the reconstructed polygons show some obvious errors including dislocation, overlap, and intersection.

In our future work, we plan to focus more on fine-grained roof structures. Although we have made progress in extracting internal feature lines within building roofs, the lack of adequate vector datasets

depicting building roof structures has led us to primarily discuss the extraction of external building outlines. We aim to build a comprehensive building dataset that includes information on vectorized and fine-grained building roof structures. Additionally, we intend to integrate data sources such as DSM or point clouds to facilitate the automatic generation of Level of Detail 2 (LOD2) 3D building models.

6. Conclusion

In this paper, we propose Line2Poly, an end-to-end method designed for extracting polygonal building footprints from remote sensing images. Unlike existing segmentation-based or vertex-based methods, Line2Poly adopts feature lines as geometric primitives and assembles them directly into building polygons with a level of detail comparable to manual delineation. The Line2Poly framework ensures the inherent regularity of predicted results, obviating the need for post-regularization steps. We have devised a two-stage strategy to fully leverage the strengths of both CNN and transformer architectures. Specifically, we utilize a CNN-based PLG module to extract preliminary feature lines as the initial positional queries for the second-stage transformer-based ALE module, which ensures the network's ability to capture long-range information while preserving convergence capability. In addition, the learnable PTR module adeptly handles instances of imperfect feature line extraction and infers adjacency relationships among discrete feature lines based on feature information. Extensive experiments on publicly available datasets underscore Line2Poly's remarkable performance in feature line extraction and instance-level building detection tasks, highlighting its capacity to generate individual polygonal buildings that align with human delineation, thus catering to the demands of real-world applications.

CRediT authorship contribution statement

Shiqing Wei: Conceptualization, Methodology, Software, Validation, Writing – original draft, Formal analysis. **Tao Zhang:** Data curation, Writing – original draft, Software. **Dawen Yu:** Visualization, Investigation. **Shunping Ji:** Conceptualization, Writing – review & editing, Supervision, Funding acquisition, Resources. **Yongjun Zhang:** Funding acquisition, Writing – review & editing. **Jianya Gong:** Resources, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (grant No. 42171430) and the State Key Program of the National Natural Science Foundation of China (grant No. 42030102).

References

Acuna, D., Ling, H., Kar, A., Fidler, S., 2018. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In: In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 859–868.

B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, "Multi-Task Learning for Segmentation of Building Footprints with Deep Neural Networks," 2017.

D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 9157-9166.

Boo, G., et al., 2022. High-resolution population estimation using household survey data and building footprints. Nature Communications 13 (1), 1–10.

Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 6, 679–698.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. In: European Conference on Computer Vision. Springer, pp. 213–229.

Castrejon, L., Kundu, K., Urtasun, R., Fidler, S., 2017. Annotating object instances with a polygon-rnn. In: In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5230–5238.

Chan, T.F., Vese, L.A., 2001. Active contours without edges. IEEE Transactions on Image Processing 10 (2), 266–277.

Chen, S., Ogawa, Y., Zhao, C., Sekimoto, Y., 2023. Large-scale individual building extraction from open-source satellite imagery via super-resolution-based instance segmentation approach. ISPRS Journal of Photogrammetry and Remote Sensing 195, 129–152.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2018. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (4), 834–848.

Chen, X., Qiu, C., Guo, W., Yu, A., Tong, X., Schmitt, M., 2022. Multiscale feature learning by transformer for building extraction from satellite images. IEEE Geoscience and Remote Sensing Letters 19, 1–5.

Chen, S., Shi, W., Zhou, M., Zhang, M., Xuan, Z., 2021. CGSANet: A Contour-Guided and Local Structure-Aware Encoder–Decoder Network for Accurate Building Extraction From Very High-Resolution Remote Sensing Imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 15, 1526–1542.

Chen, Q., Wang, L., Waslander, S.L., Liu, X., 2020. An end-to-end shape modeling framework for vectorized building outline generation from aerial images. ISPRS Journal of Photogrammetry and Remote Sensing 170, 114–126.

Cuturi, M., 2013. Sinkhorn distances: Lightspeed computation of optimal transport. Advances in Neural Information Processing Systems 26.

Dai, X., Gong, H., Wu, S., Yuan, X., Yi, M., 2022. Fully convolutional line parsing. Neurocomputing 506, 1–11.

Denis, P., Elder, J.H., Estrada, F.J., 2008. Efficient edge-based methods for estimating manhattan frames in urban imagery. In: Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12–18, 2008, Proceedings, Part II 10. Springer, pp. 197–210.

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.

K. Duan, L. Xie, H. Qi, S. Bai, Q. Huang, and Q. Tian, "Location-sensitive visual recognition with cross-iou loss," *arXiv preprint arXiv:2104.04899,* 2021.

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

He, K., Gkioxari, G., Dollar, P., Girshick, R., 2017. "Mask R-CNN," *IEEE Transactions on Pattern Analysis & Machine Intelligence,* vol. PP 99, 1.

P. V. Hough, "Method and means for recognizing complex patterns," ed: Google Patents, 1962.

Huang, W., Tang, H., Xu, P., 2021. OEC-RNN: Object-oriented delineation of rooftops with edges and corners using the recurrent neural network from the aerial images. IEEE Transactions on Geoscience and Remote Sensing 60, 1–12.

Huang, W., Liu, Z., Tang, H., Ge, J., 2021. Sequentially Delineation of Rooftops with Holes from VHR Aerial Images Using a Convolutional Recurrent Neural Network. Remote Sensing 13 (21), 4271.

Huang, K., Wang, Y., Zhou, Z., Ding, T., Gao, S., Ma, Y., 2018. Learning to parse wireframes in images of man-made environments. In: In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 626–635.

Ji, S., Wei, S., Lu, M., 2018. Fully Convolutional Networks for Multisource Building Extraction From an Open Aerial and Satellite Imagery Data Set. IEEE Transactions on Geoscience and Remote Sensing 99, 1–13. https://doi.org/10.1109/TGRS.2018.2858817.

Ji, S., Wei, S., Lu, M., 2018. A scale robust convolutional neural network for automatic building extraction from aerial and satellite imagery. International Journal of Remote Sensing 1–15. https://doi.org/10.1080/01431161.2018.1528024.

Jiang, F., Chen, J., Ji, S., 2021. Panoramic visual-inertial SLAM tightly coupled with a wheel encoder. ISPRS Journal of Photogrammetry and Remote Sensing 182, 96–111.

Li, Z., Wegner, J.D., Lucchi, A., 2019. Topological map extraction from overhead images. In: In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1715–1724.

Liang, J., Homayounfar, N., Ma, W.-C., Xiong, Y., Hu, R., Urtasun, R., 2020. Polytransform: Deep polygon transformer for instance segmentation. In: In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9131–9140.

Ling, H., Gao, J., Kar, A., Chen, W., Fidler, S., 2019. Fast interactive object annotation with curve-gcn. In: In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5257–5266.

Liu, Z., et al., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022.

Liu, Z., Liew, J.H., Chen, X., Feng, J., "dance, 2021. A Deep Attentive Contour Model for Efficient Instance Segmentation,". In: In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 345–354.

Liu, Z., Tang, H., Huang, W., 2022. Building Outline Delineation From VHR Remote Sensing Images Using the Convolutional Recurrent Neural Network Embedded With Line Segment Information. IEEE Transactions on Geoscience and Remote Sensing 60, 1–13.

S. Liu *et al.*, "Dab-detr: Dynamic anchor boxes are better queries for detr," *arXiv preprint arXiv:2201.12329,* 2022.
Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. Computer Vision and Pattern Recognition 3431–3440.
Lu, D., Mausel, P., Brondizio, E., Moran, E., 2004. Change detection techniques. International Journal of Remote Sensing 25 (12), 2365–2401.
M. Luo, S. Ji, and S. Wei, "A diverse large-scale building dataset and a novel plug-and-play domain generalization method for building extraction," *arXiv preprint arXiv: 2208.10004,* 2022.
N. Nauata and Y. Furukawa, "Vectorizing World Buildings: Planar Graph Reconstruction by Primitive Detection and Relationship Inference," Cham, 2020: Springer International Publishing, in Computer Vision – ECCV 2020, pp. 711-726.
Nauata, N., Furukawa, Y., 2020. Vectorizing world buildings: Planar graph reconstruction by primitive detection and relationship inference. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16. Springer, pp. 711–726.
Osher, S., Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics 79 (1), 12–49.
Peng, S., Jiang, W., Pi, H., Li, X., Bao, H., Zhou, X., 2020. Deep Snake for Real-Time Instance Segmentation. In: In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8533–8542.
Ronneberger, O., Fischer, P., Brox, T., October 2015. U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Munich, Germany, 5–9. Springer, pp. 234–241.
Sinkhorn, R., Knopp, P., 1967. Concerning nonnegative matrices and doubly stochastic matrices. Pacific Journal of Mathematics 21 (2), 343–348.
S. Stekovic, M. Rad, F. Fraundorfer, and V. Lepetit, "Montefloor: Extending mcts for reconstructing accurate large-scale floor plans," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16034-16043.
R. Stewart, M. Andriluka, and A. Y. Ng, "End-to-end people detection in crowded scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2325-2333.
Sun, Z., Zhou, W., Ding, C., Xia, M., 2022. Multi-resolution transformer network for building and road segmentation of remote sensing image. ISPRS International Journal of Geo-Information 11 (3), 165.
Vaswani, A., et al., 2017. Attention is all you need. Advances in Neural Information Processing Systems 30.
Von Gioi, R.G., Jakubowicz, J., Morel, J.-M., Randall, G., 2008. LSD: A fast line segment detector with a false detection control. IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (4), 722–732.
Wang, L., et al., 2022. UNetFormer: A UNet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. ISPRS Journal of Photogrammetry and Remote Sensing 190, 196–214.
Wang, L., Fang, S., Meng, X., Li, R., 2022. Building extraction with vision transformer. IEEE Transactions on Geoscience and Remote Sensing 60, 1–11.
Wang, X., Kong, T., Shen, C., Jiang, Y., Li, L., 2020. Solo: Segmenting objects by locations. In: European Conference on Computer Vision. Springer, pp. 649–665.
Wei, S., Ji, S., 2021. Graph Convolutional Networks for the Automated Production of Building Vector Maps From Aerial Images. IEEE Transactions on Geoscience and Remote Sensing.
Wei, S., Ji, S., Lu, M., 2019. Toward Automatic Building Footprint Delineation From Aerial Images Using CNN and Regularization. In: IEEE Transactions on Geoscience and Remote Sensing, pp. 1–12. https://doi.org/10.1109/TGRS.2019.2954461.
Wei, F., Sun, X., Li, H., Wang, J., Lin, S., 2020. "Point-Set Anchors for Object Detection. Instance Segmentation and Pose Estimation," *arXiv Preprint arXiv:2007.02846*.
Wei, S., Zhang, T., Ji, S., 2021. A Concentric Loop Convolutional Neural Network for Manual Delineation-Level Building Boundary Segmentation From Remote-Sensing Images. IEEE Transactions on Geoscience and Remote Sensing 60, 1–11.
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.
Wu, T., Hu, Y., Peng, L., Chen, R., 2020. Improved Anchor-Free Instance Segmentation for Building Extraction from High-Resolution Remote Sensing Images. Remote Sensing 12 (18), 2910.
Xiao, X., Guo, W., Chen, R., Hui, Y., Wang, J., Zhao, H., 2022. A swin transformer-based encoding booster integrated in u-shaped network for building extraction. Remote Sensing 14 (11), 2611.
Xie, E., et al., 2020. Polarmask: Single shot instance segmentation with polar representation. In: In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12193–12202.
Xu, Y., Xu, W., Cheung, D., Tu, Z., 2021. Line segment detection using transformers without edges. In: In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4257–4266.
Xue, N., et al., 2020. Holistically-attracted wireframe parsing. In: In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2788–2797.
N. Xue, S. Bai, F. Wang, G.-S. Xia, T. Wu, and L. Zhang, "Learning attraction field representation for robust line segment detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1595-1603.
Xue, N., Xia, G.-S., Bai, X., Zhang, L., Shen, W., 2017. Anisotropic-scale junction detection and matching for indoor images. IEEE Transactions on Image Processing 27 (1), 78–91.
Yeh, A.G., 1999. Urban planning and GIS. Geographical Information Systems 2 (877–888), 1.
J. Yi, P. Wu, B. Liu, Q. Huang, H. Qu, and D. Metaxas, "Oriented object detection in aerial images with box boundary-aware vectors," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 2150-2159.
F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2403-2412.
Yuan, J., 2017. "Learning Building Extraction in Aerial Scenes with Convolutional Networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence,* vol. PP 99, 1.
F. Zhang, N. Nauata, and Y. Furukawa, "Conv-mpn: Convolutional message passing neural network for structured outdoor architecture reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2798-2807.
Zhang, T., Wei, S., Ji, S., 2022. E2ec: An end-to-end contour-based method for high-quality high-speed instance segmentation. In: In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4443–4452.
W. Zhao, C. Persello, and A. Stein, "Building instance segmentation and boundary regularization from high-resolution remote sensing images," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, 2020: IEEE, pp. 3916-3919.
Zhao, K., Kang, J., Jung, J., Sohn, G., 2018. Building Extraction from Satellite Images Using Mask R-CNN with Building Boundary Regularization. In: In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 247–251.
Zhao, W., Persello, C., Stein, A., 2021. Building outline delineation: From aerial images to polygons with an improved end-to-end learning framework. ISPRS Journal of Photogrammetry and Remote Sensing 175, 119–131.
Zhao, W., Persello, C., Stein, A., 2022/05/01/ 2022,. Extracting planar roof structures from very high resolution images using graph neural networks. ISPRS Journal of Photogrammetry and Remote Sensing 187, 34–45. https://doi.org/10.1016/j.isprsjprs.2022.02.022.
Zhou, Y., Qi, H., Ma, Y., 2019. End-to-end wireframe parsing. In: In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 962–971.
X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159,* 2020.
C. Zhu, X. Zhang, Y. Li, L. Qiu, K. Han, and X. Han, "SharpContour: A Contour-based Boundary Refinement Approach for Efficient and Accurate Instance Segmentation," *arXiv preprint arXiv:2203.13312,* 2022.
Zhu, Q., Liao, C., Hu, H., Mei, X., Li, H., 2020. MAP-Net: Multiple attending path neural network for building footprint extraction from remote sensed imagery. IEEE Transactions on Geoscience and Remote Sensing 59 (7), 6169–6181.
Zorzi, S., Bazrafkan, S., Habenschuss, S., Fraundorfer, F., 2021. PolyWorld: Polygonal Building Extraction with Graph Neural Networks in Satellite Images.
S. Zorzi, K. Bittner, and F. Fraundorfer, "Machine-learned regularization and polygonization of building segmentation masks," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021: IEEE, pp. 3098-3105.