Re:PolyWorld - A Graph Neural Network for Polygonal Scene Parsing

Stefano Zorzi
Graz University of Technology
stefano.zorzi@icg.tugraz.at

Friedrich Fraundorfer
Graz University of Technology
fraundorfer@icg.tugraz.at

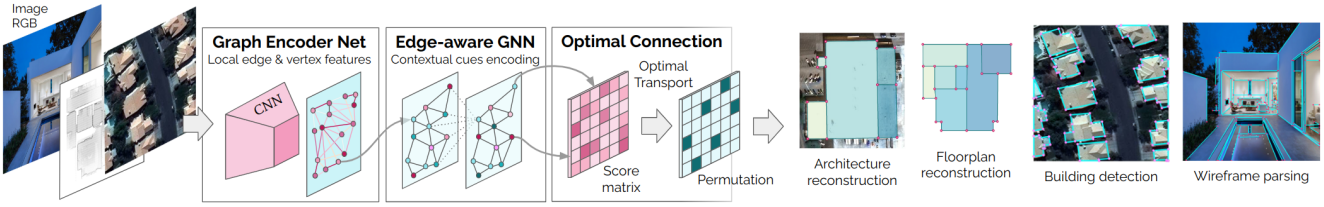

Figure 1: PolyWorld Remastered (Re:PolyWorld) extracts local vertex and edge features from an intensity image, and embeds global information about the scene by using an Edge-aware GNN. The connections between vertices are generated by solving a differentiable optimal transport problem. By redefining the representation of the polygonal scene, this method becomes a generalized approach that can be applied to a variety of tasks and problem settings.

Abstract

While most state-of-the-art instance segmentation methods produce pixel-wise segmentation masks, numerous applications demand precise vector polygons of detected objects instead of rasterized output. This paper proposes Re:PolyWorld as a remastered and improved version of PolyWorld, a neural network that extracts object vertices from an image and connects them optimally to generate precise polygons. The objective of this work was to overcome weaknesses and shortcomings of the original model, as well as introducing an improved polygonal representation to obtain a general-purpose method for polygon extraction in images. The architecture has been redesigned to not only exploit vertex features, but to also make use of the visual appearance of edges. To this end, an edge-aware Graph Neural Network predicts the connection strength between each pair of vertices, which is further used to compute the assignment by solving a differentiable optimal transport problem. The proposed redefinition of the polygonal scene turns the method into a powerful generalized approach that can be applied to a large variety of tasks and problem settings, such as building extraction, floorplan reconstruction and even wireframe parsing. Re:PolyWorld not only outperforms the original model on building extraction in aerial images, thanks to the proposed joint analysis of vertices and edges, but also beats the state-of-the-art in multiple other domains.

1. Introduction

The detection of polygonal objects and the generation of vector annotations is an important topic in numerous computer vision applications, such as building extraction in airborne imagery, floorplan reconstruction, scene understanding, as well as wireframe parsing. In all these applications, the visual quality of the final polygonization is a highly desired feature that modern machine learning approaches aim to achieve. On one hand, vector annotations with redundant vertices often appear irregular, and exhibit artifacts or curved corners. On the other hand, oversimplified polygons are often not able to capture geometric details of the object.

In this paper, we propose Re:PolyWorld (Figure 1) as a remastered version of PolyWorld [30], a neural network for building detection and polygonization in airborne imagery. The original model extracts positions and visual descriptors of building corners using a Convolutional Neural Network (CNN) and generates polygons by evaluating which connections between vertices are valid. This procedure identifies the optimal pairing of detected vertex descriptors, requiring each corner to be linked with its respective subsequent vertex in the polygon. Connections between polygon vertices can be represented by a permutation matrix obtained as the solution of a linear sum assignment problem. PolyWorld is fast, accurate, end-to-end trained, and generates high quality building polygons ready to be used in cartographic applications. However, the permutational representation of the scene is not applicable to other problem settings and applications, and the model only reasons

based on local visual descriptors of vertices.

Motivated by these drawbacks of PolyWorld, our remastered model makes the following contributions:

- We propose a novel generalized representation of the polygonal scene, that can be applied to numerous tasks and problem settings with diverse image data sets.
- We modify the PolyWorld architecture in order to not only rely on local vertex information, but also exploit visual edge features by introducing a novel edge-aware attention mechanism.
- We evaluate our model on a variety of data sets and different tasks; including building extraction, floorplan reconstruction, and even wireframe parsing.

2. Related Work

Building detection and polygonization is generally performed in a divide-and-conquer manner, by applying complex post-processing vectorizations and polygon simplification algorithms to the pixel-wise prediction of instance segmentation networks [26, 2, 32, 31]. PolyMapper [10] detects each building by producing a bounding box, and subsequently applies a RNN to predict building and road vertices one by one. Frame Field Learning (FFL) [6, 5] is currently one of the most effective approaches for extracting building polygons. It produces a vector field that encodes useful boundary information and a corresponding segmentation mask. The contour is optimized by an Active Skeleton Model in a post-processing step.

Floorplan and Architecture reconstruction methods are also relying on image segmentation as initial step. FloorSP [1], receives aligned panorama RGBD scans as input, identifies room segments by using Mask R-CNN [7], and reconstructs a floorplan graph by solving an optimization problem, resulting in multiple polygonal loops. Finally, the loops are merged into a 2D graph through post-processing heuristics. Montefloor [18] also detects room segments using Mask R-CNN, and generates room proposals by polygonizing the segments with different hyperparameters. To choose the correct room proposals, the authors use Monte Carlo Tree Search coupled with a proper objective function, and refine the proposed room shapes to fit the input density map. HEAT [9] proposes an attention-based neural network for structured reconstruction, which detects corners and classifies edge candidates between corners using a holistic edge classification architecture.

Wireframe parsing networks are inspired by state-of-the-art deep learning models used in human pose estimation [15, 19, 28]. This has led to the development of a wireframe parsing algorithm called L-CNN [29] that, unlike other methods, utilizes a sampling scheme to generate line segment proposals based on predicted junctions, and a line segment verification module to classify them. However, L-CNN incurs high computational costs due to the large number of proposals it generates. Moreover, ignoring line segment information in the proposal stage may not fully leverage the deep learning pipeline for better performance. Even though not fully end-to-end trainable, attraction field map (AFM) based approaches [22, 23] reach outstanding performance in line segment detection without exploiting junction information during learning. More recently, HAWP [24, 25], similarly to L-CNN, performs line and junctions proposal generation and proposal verification. It achieves state-of-the-art performance by introducing a novel line segment prediction approach for more accurate parsing.

3. PolyWorld Remastered

A Remastered Polygonal Scene: Similar to PolyWorld, this work aims to represent polygons in an image as a set of vertices, and by a corresponding matrix that encodes connection between them. In the original setup of PolyWorld, this matrix is a permutation matrix, which allows to train the neural network by minimizing an optimal transport loss in an end-to-end fashion. Thereby, the model is enforced to predict strong edges and ultimately allows the generation of precise polygons. However, that approach suffers from a major limitation: if an image contains polygons with shared vertices, the edges cannot longer be represented by a permutation.

To overcome this problem, we propose a novel representation of the polygonal scene, where every vertex $\mathbf{v}_i$ is represented as a set of K vertex instances $\mathbf{v}_i = \{\mathbf{v}_i^1, \mathbf{v}_i^2, ..., \mathbf{v}_i^K\}$, which allows the vertex $\mathbf{v}_i$ to have at most K different edges. Using this multi-instance representation of vertices, we describe the objects in the scene by clockwise oriented polygons, with their edges encoded by an adjacency matrix $\mathbf{A}$, as shown in Figure 2. Each vertex is associated with a specific row of the adjacency matrix, thereby indicating the connections of its instances. Since redundant vertex instances are considered self-loops and are assigned along the diagonal of the adjacency matrix, each row and column of $\mathbf{A}$ sums to K.

Expressing the vertices as their single instance components, the matrix $\mathbf{A}$ can be expanded and represented as a permutation matrix $\mathbf{P}$, where each row is assigned to a specific vertex instance $\mathbf{v}_i^j$ and indicates the next clockwise connection. Two observations can be made about the relationship between $\mathbf{A}$ and $\mathbf{P}$:

1 The entry $\mathbf{A}_{i,j}$, representing the number of connections between vertices $\mathbf{v}_i$ and $\mathbf{v}_j$, is equal to the sum of the instance elements that are connecting $\mathbf{v}_i$ to $\mathbf{v}_j$ in the permutation matrix $\mathbf{P}$.

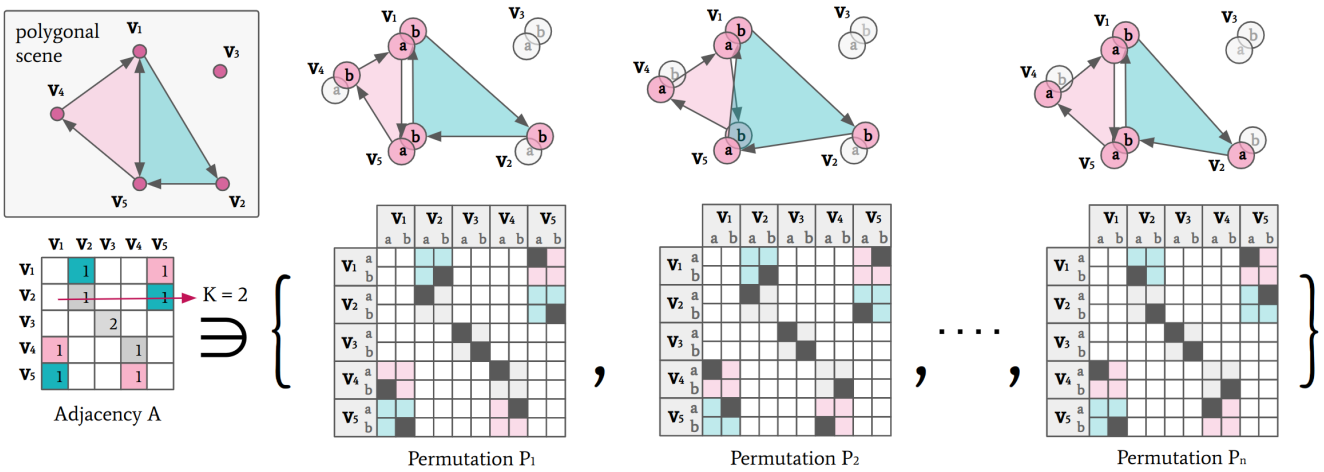

Figure 2: Example of a polygonal representation in Re:PolyWorld. Each vertex in the scene has K instances that allow to have multiple connections (in this example $K = 2$). The connections between vertices can therefore be represented by an adjacency matrix $\mathbf{A}$, where rows and columns sum to K (left side), since unconnected instances can be assigned to the diagonal. The adjacency matrix can be expanded to a permutation matrix that encodes the connection of each vertex instance (right side). In this example, the two instances of each vertex are indicated by the symbols a and b. Note that the permutation matrix is not unique, i.e. multiple equivalent representations of the polygons $[\mathbf{v}_1 \rightarrow \mathbf{v}_5 \rightarrow \mathbf{v}_4]$ and $[\mathbf{v}_1 \rightarrow \mathbf{v}_2 \rightarrow \mathbf{v}_5]$ are possible by using different vertex instance combinations.

2 $\mathbf{P}$ is not unique. In fact, multiple equivalent permutation matrices can represent $\mathbf{A}$, and ultimately the same polygonal scene, with different vertex instance combinations.

Architecture Introduction: Re:PolyWorld segments polygonal objects in an input image, by generating a set of 2D coordinates (representing the vertices) and a permutation matrix (encoding the connection between them). The model detects salient vertices in the image and embeds their visual features in descriptors. These representations are used to match vertices and, ultimately, to generate a permutation matrix.

Although the primary goal of the proposed method is to detect and match polygon vertices, also the visual appearance of edges are powerful features to disambiguate vertex connections. When asked to find the next connection of a vertex, humans visually follow the edge in an image, until the subsequent node is reached. It is evident that edge information can resolve ambiguity in node connectivity in presence of self-similar vertices; moreover, discontinuities of the edge can penalize a match.

Motivated by this consideration, we present an edge-aware neural network composed of three blocks: a Graph Encoder Network (GEN) that detects salient vertices in the image and generates a visual descriptor for both the nodes and the edges; an Edge-Aware Graph Neural Network (EA-GNN) that reasons about the graph and updates node and edge representations; and an Optimal Connection Network (OCN) that performs vertex matching in an optimal way. An overview of Remastered PolyWorld is shown in Figure 3.

3.1. Graph Encoder Network

The first module of PolyWorld Remastered is a neural network for graph encoding that receives an intensity image $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ as input and encodes it in a feature map $\mathbf{F} \in \mathbb{R}^{D\times H\times W}$ using a CNN backbone. The module predicts a vertex detection map $\mathbf{Y} \in \mathbb{R}^{H\times W}$ by propagating $\mathbf{F}$ through a pixel-wise projection, and generates a set of N *2D-positions* p_i representing the top-N detected peaks after filtering $\mathbf{Y}$ with a Non-Maximum Suppression layer (NMS).

Vertex and edge encodings: The position p_i of a detected vertex, is further exploited to fetch visual descriptors from the feature map $\mathbf{F}$: the *vertex descriptor* $\mathbf{d}_i \in \mathbb{R}^D$ is simply obtained by fetching the feature map $\mathbf{F}$ at the position p_i, while the *edge descriptor* $\mathbf{e}_{i\rightarrow j} \in \mathbb{R}^D$ is computed as follows:

$$\mathbf{e}_{i\rightarrow j} = \text{MLP}_{edge}\left(\left[\Lambda(\mathbf{d}^1_{i\rightarrow j})||\Lambda(\mathbf{d}^2_{i\rightarrow j})||\cdots||\Lambda(\mathbf{d}^M_{i\rightarrow j})\right]\right) \quad (1)$$

where $\{\mathbf{d}^1_{i\rightarrow j}, \mathbf{d}^2_{i\rightarrow j}, \cdots, \mathbf{d}^M_{i\rightarrow j}\}$ are M descriptors obtained by linearly sampling the feature map $\mathbf{F}$ from position p_i to p_j, therefore $\mathbf{d}_i = \mathbf{d}^1_{i\rightarrow j}$ and $\mathbf{d}_j = \mathbf{d}^M_{i\rightarrow j}$. Λ and MLP_{edge} are a linear projection and a Multi-Layer Perceptron, respectively, used to map the input vector into a compressed repre-

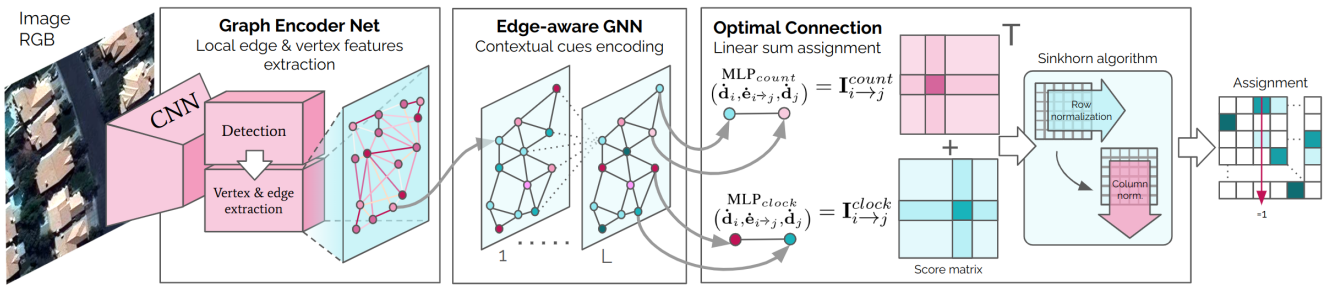

Figure 3: Overview of Re:PolyWorld. The Graph Encoder Network extracts a set of local vertex and edge features from the intensity image. An edge-aware Graph Neural Network embeds global information of the scene by analyzing the extracted vertex and edge representations. The Optimal Connection Network generates connections between vertices, encoded in a permutation matrix, by solving a linear sum assignment problem.

sentation: Λ compresses the single vertex descriptor, while MLP_{edge} receives the concatenation $[\cdot||\cdot]$ of the sampled descriptors and generates a compact representation of the path $p_i \rightarrow p_j$. The chosen hyperparameters, and the dimension of the intermediate tensors, are reported in the supplementary material.

3.2. Graph Neural Network

Local vertex and edge features, like visual appearance, pose, vertex angles, and edge discontinuities are encoded by the backbone CNN. However, enriching these representations with contextual cues and priors computed by aggregating long-range features are extremely useful to describe vertices and edges in a more distinctive and refined way. Consequently, we design the next module as an edge-aware Attentional Graph Neural Network.

Given the initial vertex and edge features extracted by the backbone, the module computes *vertex matching descriptors* $\dot{\mathbf{d}}_i \in \mathbb{R}^D$, and *edge matching descriptors* $\dot{\mathbf{e}}_{i\to j} \in \mathbb{R}^D$ by allowing information propagation through the features.

Edge-Aware Self-Attention Network: Since we desire to update the local representation of the elements detected in the image through a global flow of information, we consider a fully-connected graph where the nodes are represented by the detected vertices $\mathbf{d}_i$ and the edges carry the attribute $\mathbf{e}_{i\to j}$. The information is propagated along the edges via a message passing formulation. The module consists of L layers, where each of them computes a globally refined representation of all graph elements by aggregating messages across all given edges in a concurrent way.

The state $\mathbf{d}_i^{(\ell)}$, representing the i-th vertex at layer ℓ, is updated as follows:

$$\mathbf{d}_i^{(\ell+1)} = \mathbf{d}_i^{(\ell)} + \text{MLP}\left(\left[\mathbf{d}_i^{(\ell)}||\mathbf{m}_i^{(\ell)}\right]\right) \tag{2}$$

where $\mathbf{m}_i^{(\ell)}$ is the message relative to the i-th vertex obtained by aggregating all current node states.

Similarly, the attributes of the edge $\mathbf{e}_{i\to j}^{(\ell)}$ connecting node i and node j are locally updated using the global information provided by the messages $\mathbf{m}_i^{(\ell)}$ and $\mathbf{m}_j^{(\ell)}$:

$$\mathbf{e}_{i\to j}^{(\ell+1)} = \mathbf{e}_{i\to j}^{(\ell)} + \text{MLP}\left(\left[\mathbf{m}_i^{(\ell)}||\mathbf{e}_{i\to j}^{(\ell)}||\mathbf{m}_j^{(\ell)}\right]\right) \tag{3}$$

The embeddings received by the first layer are the visual descriptors extracted by the backbone: $\mathbf{d}_i = \mathbf{d}_i^{(0)}$, $\mathbf{e}_{i\to j} = \mathbf{e}_{i\to j}^{(0)}$; while the final states are used by the OCN as matching descriptors: $\dot{\mathbf{d}}_i = \mathbf{d}_i^{(L)}$, $\dot{\mathbf{e}}_{i\to j} = \mathbf{e}_{i\to j}^{(L)}$.

Attentional aggregation: The aggregation of the message $\mathbf{m}_i$ is performed by a self-attention mechanism [20], that employs the linear projections Q, K, V to produce queries $\mathbf{q}_i = Q(\mathbf{d}_i)$, keys $\mathbf{k}_j = K(\mathbf{d}_j)$, and values $\mathbf{v}_j = V(\mathbf{d}_j)$. The result is calculated as the weighted sum of the values:

$$\mathbf{m}_i = \sum_{j=1}^{N} \alpha_{ij}\mathbf{v}_j \tag{4}$$

where the attentional weight α_{ij} is computed as the $\mathrm{softmax}$ over the similarities between queries and keys, scaled by a factor σ_{ij} representing the connectivity between the nodes:

$$\alpha_{ij} = \text{softmax}_j\left(\sigma_{ij}\mathbf{q}_i^\top \mathbf{k}_j\right) \tag{5}$$

The scale factor σ_{ij} is related to the edge attributes, and is computed by linearly projecting the edge features: $\sigma_{ij} = E(\mathbf{e}_{i\to j})$. The expressiveness of the network is augmented by implementing multi-headed attention layers [20], each of which has its own parameters for the projections.

Positional refinement: In some applications, like building extraction or floor-plan reconstruction, it is crucial to generate realistic polygons that typically require having realistic shapes and 90-degrees corner angles. Instead of relying on complex post processing steps of polygonal refinement, it is possible to efficiently estimate a *positional offset*

$\mathbf{o}_i \in [-r, r]^{2\times N}$ for every vertex by exploiting the result of the GNN aggregation:

$$\mathbf{o}_i = \text{MLP}_{offset}(\dot{\mathbf{d}}_i) \tag{6}$$

We obtain refined positions by combining the initial positions with offsets constrained on having a maximum radius of r: $\hat{\mathbf{p}}_i = \mathbf{p}_i + \mathbf{o}_i$. The offsets, depending on the application, can be learned by minimizing a proper mixture of regularization losses [30] (see Section 3.4).

3.3. Optimal Connection Network

Once the visual descriptors of the graph elements are enriched with contextual cues and global information, an Optimal Connection Network exploits these representations to connect the detected vertices by generating a permutation matrix $\mathbf{P} \in [0, 1]^{KN\times KN}$. The module evaluates the connection strength between each pair of vertex instances by computing a score matrix $\mathbf{S} \in \mathbb{R}^{KN\times KN}$, and estimates the optimal assignment by minimizing the overall score $\sum_{ij} \mathbf{P}_{ij}\mathbf{S}_{ij}$.

Given the matching descriptors $\dot{\mathbf{d}}_i$, $\dot{\mathbf{d}}_j$, and $\dot{\mathbf{e}}_{i\to j}$, the network evaluates whether the i-th vertex embeds instances clockwise-connected to the j-th vertex by calculating intra-vertex scores $\mathbf{I}^{clock}_{i\to j} \in \mathbb{R}^{K\times K}$:

$$\mathbf{I}^{clock}_{i\to j} = \text{MLP}_{clock}\left(\left[\dot{\mathbf{d}}_i||\dot{\mathbf{e}}_{i\to j}||\dot{\mathbf{d}}_j\right]\right) \tag{7}$$

The overall clockwise score matrix $\mathbf{S}_{clock}$ can therefore be assembled as a block matrix of the single intra-vertex score matrices:

$$\mathbf{S}_{clock} = \begin{bmatrix} \mathbf{I}^{clock}_{1\to 1} & \mathbf{I}^{clock}_{1\to 2} & \cdots & \mathbf{I}^{clock}_{1\to N} \\ \mathbf{I}^{clock}_{2\to 1} & \mathbf{I}^{clock}_{2\to 2} & \cdots & \mathbf{I}^{clock}_{2\to N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}^{clock}_{N\to 1} & \mathbf{I}^{clock}_{N\to 2} & \cdots & \mathbf{I}^{clock}_{N\to N} \end{bmatrix} \tag{8}$$

In order to establish a double-path consistency[30] between the clockwise and the counter-clockwise path of vertices, we compute the score matrix $\mathbf{S}_{count}$ using a second MLP:

$$\mathbf{I}^{count}_{i\to j} = \text{MLP}_{count}\left(\left[\dot{\mathbf{d}}_i||\dot{\mathbf{e}}_{i\to j}||\dot{\mathbf{d}}_j\right]\right) \tag{9}$$

Since the permutation matrix of the clockwise oriented polygons is the transpose of the counter-clockwise permutation matrix [30], the final score matrix can be computed as the ensemble of the two vertex paths: $\mathbf{S} = \mathbf{S}_{count} + \mathbf{S}^{\top}_{count}$. The use of double-path consistency guarantees improved reliability of scores, resulting in stronger matches and ultimately leading to more precise polygon representations.

Finally, we compute the optimal assignment $\mathbf{P}$, given the score matrix $\mathbf{S}$, by applying the Sinkhorn algorithm [3, 16, 17], which is a differentiable version of the Hungarian optimization [14] used to solve linear sum assignment problems. The Sinkhorn algorithm consists of an iterative normalization of the rows and columns of $\exp(\mathbf{S})$, therefore, it can be parallelized and efficiently computed on a GPU.

3.4. Training Losses

The general training objectives of Re:PolyWorld coincide with the original model [30]. Hence, our explanation is confined to offering a brief overview. The network is trained end-to-end in a fully supervised manner with three kinds of objectives: detection, matching, and refinement.

Detection & Matching: The first two losses are essential for the correct training of the model. The network learns to detect vertices by minimizing the weighted cross-entropy loss $\mathcal{L}_{det}$ between the predicted detection map $\mathbf{Y}$ and the ground truth vertex occupancy grid $\bar{\mathbf{Y}}$. Moreover, it learns to generate vertex connections by optimizing the cross-entropy loss $\mathcal{L}_{match}$ between the assignment $\mathbf{P}$ and the ground truth permutation $\bar{\mathbf{P}}$.

Refinement: Depending on the application, it can be beneficial to learn a refinement offset $\mathbf{o}_i$ in order to optimize the final polygon shapes. To this end, we use a combination of the angle loss $\mathcal{L}_{ang}$ and segmentation loss $\mathcal{L}_{seg}$ suggested in [30]. These training objectives encourage the network to produce polygons with the typical 90-degree corners, visually desired in building footprint extraction or floor-plan reconstruction, while, at the same time, improving segmentation scores.

3.5. Training Procedure

Vertex synchronization: During training, we sample N vertices with the NMS layer, ensuring N always being greater than the number of keypoints present in an image annotation. Therefore, the ground truth vertices, with their respective permutation matrix, must be synchronized with the detection, in order to train the matching path. The simplest approach is to associate every ground truth vertex to the nearest predicted vertex in terms of euclidean distance. The redundant predicted vertices are assigned to the diagonal of the permutation matrix.

Permutation guidance: Since Re:PolyWorld exploits multiple vertex instances ($K > 1$) to generalize the representation of the polygonal scene, the ground truth connections are encoded by an adjacency matrix $\bar{\mathbf{A}}$. As a result, the matching loss $\mathcal{L}_{match}$ cannot be directly computed, since the ground truth permutation matrix $\bar{\mathbf{P}}$ is not uniquely defined [2].

In order to overcome this problem, and guide the network to learn the correct representation, we compute the optimal ground truth permutation matrix $\bar{\mathbf{P}}^*$ that maximizes

Figure 4: Examples of building extraction and polygonization on the CrowdAI [13] test dataset by using Frame Field Learning [6] with ACM polygonization, and PolyWorld Remastered.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR	AR_{50}	AR_{75}	AR_S	AR_M	AR_L
Mask R-CNN [7]	41.9	67.5	48.8	12.4	58.1	51.9	47.6	70.8	55.5	18.1	65.2	63.3
PANet [12]	50.7	73.9	62.6	19.8	68.5	65.8	54.4	74.5	65.2	21.8	73.5	75.0
PolyMapper [10]	55.7	86.0	65.1	30.7	68.5	58.4	62.1	88.6	71.4	39.4	75.6	75.4
FFL, simple poly [6]	61.7	87.6	71.4	35.7	74.9	83.0	65.4	89.8	74.6	42.5	78.6	85.8
FFL, ACM poly [6]	61.3	87.4	70.6	33.9	75.1	83.1	64.9	89.4	73.9	41.2	78.7	85.9
PolyWorld [30]	63.3	88.6	70.5	37.2	83.6	87.7	75.4	93.5	83.1	52.5	88.7	95.2
Re:PolyWorld	**67.2**	**89.8**	**75.8**	**42.9**	**85.3**	**89.4**	**78.6**	**94.1**	**86.7**	**58.3**	**90.3**	**96.2**

Table 1: MS COCO [11] results on the CrowdAI test dataset [13] for building detection. The results of PolyWorld and the Remastered are calculated using the refinement offset for the vertex positions. FFL refers to the Frame Field Learning [6] method. "simple poly" refers to the Douglas–Peucker polygon simplification [4], and "ACM poly" refers to the Active Contour Model [6] polygonization method.

Method	IoU	C-IoU	MTA
FFL, simple poly [6]	84.0	30.1	48.2°
FFL, ACM poly [6]	84.1	73.7	33.5°
PolyWorld [30]	91.3	88.2	32.9°
Re:PolyWorld	**92.2**	**89.7**	**31.9°**

Table 2: *Intersection over union (IoU)*, *max tangent angle error (MTA)* [6], and *complexity aware IoU (C-IoU)* [30] results on the test-set of the CrowdAI dataset [13].

		Room		Corner		Angle	
Method	t(s)	Prec	Recall	Prec	Recall	Prec	Recall
HAWP [24]	**0.02**	0.78	0.88	0.66	0.77	0.6	0.70
LETR [21]	0.04	0.94	0.90	0.8	0.78	0.72	0.71
Floor-SP [18]	785	0.89	0.88	0.81	0.73	0.8	0.72
MonteFloor [18]	71	0.96	0.94	**0.89**	0.77	**0.86**	0.75
HEAT [9]	0.11	0.97	0.94	0.82	0.83	0.78	0.79
Re:PolyWorld	**0.02**	**0.99**	**0.97**	0.81	**0.86**	0.77	**0.82**

Table 3: Floorplan reconstruction results on the Structured3D dataset [27].

the assignment of the predicted scores $\mathbf{S}$, conditioned by the ground truth adjacency matrix $\bar{\mathbf{A}}$:

$$\begin{aligned} \bar{\mathbf{P}}^{n*} &= \max_{\bar{\mathbf{P}}^n} \sum_{i,j} \bar{\mathbf{P}}^n_{i,j} \cdot \mathbf{S}^n_{i,j} \\ \text{subject to} \quad \bar{\mathbf{A}}^n_{i,j} &= \sum_{\{k,q\}=1}^{K} \bar{\mathbf{P}}^n_{v_i^k, v_j^q} \end{aligned} \tag{10}$$

The constraint of the optimization problem enforces the relationship $\boxed{1}$ between $\bar{\mathbf{A}}$ and $\bar{\mathbf{P}}$. It is worth noting that the matrices are raised to the power of n to utilize a property of adjacency matrices: the element $\mathbf{A}^n_{i,j}$ gives the number of directed walks of length n from vertex i to vertex j. This property can therefore be used to discard solutions (from the set of possible permutations) that encode polygons with a specific number of vertices. In applications like floorplan reconstruction or building detection, we set $n = 2$ to avoid solutions representing polygons with two vertices (lines). In wireframe parsing, line detection is desired, therefore we use $n = 1$.

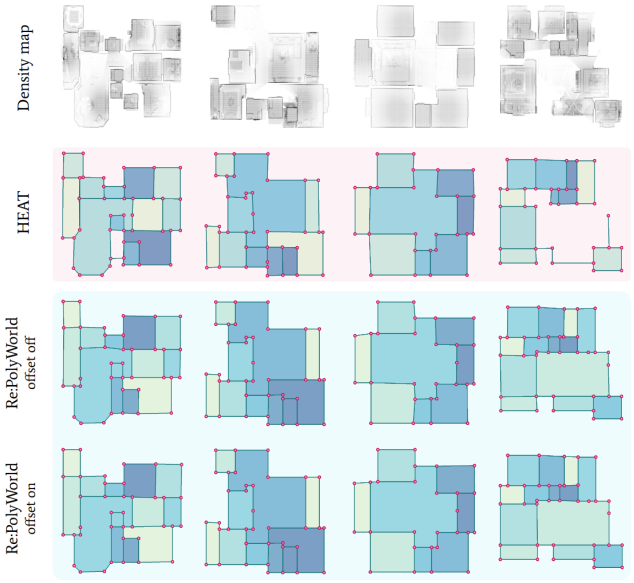

Figure 5: Floorplan reconstruction results on the S3D dataset [27] by HEAT [9] and Re:PolyWorld. "Offset off" refers to the result without using the positional refinement offset. "Offset on" refers to the full method.

Figure 6: Outdoor building reconstruction task [9]. Results obtained by HEAT [9] and Re:PolyWorld.

Training stages: Re:PolyWorld is trained in three steps. Firstly, the vertex detection path is pretrained by only using $\mathcal{L}_{det}$. Secondly, detection and matching paths are jointly trained combining $\mathcal{L}_{det} + \mathcal{L}_{match}$. Finally, the offset is learned by training the network with the full objective, if the application requires it: $\mathcal{L}_{det} + \mathcal{L}_{match} + \mathcal{L}_{ang} + \mathcal{L}_{seg}$.

4. Experiments

Building extraction: We conduct our experiments for building extraction using the CrowdAI Mapping Challenge dataset [13]. We consider these experiments an ablation study, with respect to the proposed edge-aware attention mechanism, while keeping backbone, training procedure, and hyperparameters of the network as in the original PolyWorld [30] settings. The quantitative results reported in Table 1 and 2 show the improvement in terms of MS-COCO [11] metrics and segmentation scores obtained by additionally exploiting edge information. Re:PolyWorld achieves, in fact, not only higher instance precision and recall scores, but it also improves the footprint quality as indicated by the complexity-aware IoU [30], and max tangent angle error (MTA) [6]. Qualitative results are shown in Figure 4.

Floorplan & outdoor reconstruction: To demonstrate the generalization capabilities of our model for polygons with shared vertices, we perform experiments on the S3D dataset [27] for floorplan reconstruction, and on the HEAT dataset [9] for outdoor building reconstruction. Both of these datasets require a model capable of connecting vertices to multiple nodes, making the polygon detection task challenging. However, by training Re:PolyWorld with a maximum number of vertex instances $K = 4$, we can effectively generalize to these polygonal scenes, producing precise and visually accurate polygons, as shown in Figure 5 and 6. Notably, when the refinement offset is enabled to enhance vertex positioning, the resulting polygons show a more realistic appearance and are ready for use in real-world applications without requiring expensive post-processing steps. In contrast to our model, that generates polygons directly as output, HEAT [9] predicts edges and nodes. Consequently, a conversion process is required to transform the edge and node representation into polygonal forms. Results in terms of recall and precision are reported in Table 3.

Wireframe parsing: Re:PolyWorld can also be adapted to predict image segments, or wireframes, by defining each line as a two-vertex polygon. By interpreting the polygonal scene this way, the resulting permutation matrix $\mathbf{P}$ is symmetric, and therefore the double-path consistency is not required. With these considerations, we compute the score matrix $\mathbf{S}$ by using a single MLP:

$$\mathbf{I}_{i \leftrightarrow j} = \underset{wire}{\text{MLP}}\left(\left[\dot{\mathbf{d}}_i || \dot{\mathbf{e}}_{i \to j} || \dot{\mathbf{d}}_j\right]\right) + \underset{wire}{\text{MLP}}\left(\left[\dot{\mathbf{d}}_j || \dot{\mathbf{e}}_{j \to i} || \dot{\mathbf{d}}_i\right]\right) \tag{11}$$

We tested our approach on the Wireframe parsing dataset [29, 8] using $K = 4$ as maximum number of vertex instances. In contrast to state-of-the-art wireframe parsing methods, Re:PolyWorld does not rely on filtering proposals with low confidence score, but rather on solving the optimal transport problem. This approach results in a significantly lower number of wireframe proposals in comparison to other methods, with the same order of magnitude as the number of ground truth lines (see Table 4). Qualitative results on the Wireframe parsing dataset [8] are shown in Figure 7.

Figure 7: Qualitative evaluation of wireframe detection on the Wireframe Parsing dataset [29] by HAWPv2 [25] and Re:PolyWorld. The wireframes visualized for the HAWP method have $score > 0.9$. The wireframes detected by Re:PolyWorld are obtained by solving the linear sum assignment problem.

Method	# Proposals	sAP^{10}	R^{10}	# GT lines
L-CNN [29]	22000	63.6	-	74.2
HAWPv1 [24]	4000	66.5	-	
HAWPv2 [25]	1000	**69.7**	63.1	
Re:PolyWorld	**83**	50.2	**64.6**	

Table 4: Results on the Wireframe parsing dataset [29]. The average number of ground truth lines is listed in the last column. The number of proposals of Re:PolyWorld is the average number of lines generated by solving the linear sum assignment problem.

5. Limitations

The proposed method consists of detecting keypoints in an image and connecting them correctly to create polygons in a fully-supervised manner. Training this kind of networks requires high quality vector annotations to successfully train a keypoint detector. Noisy annotations, such as those that are misaligned or missing, may inhibit the detection process from identifying all vertices, resulting in the omission of object corners during the matching phase. In this case, we experimentally noticed that the network either produces polygons with missing corners, or discards the entire object footprint. Visual examples of this undesired behaviour are shown in the supplementary material.

6. Conclusions

This paper introduces a remastered and improved version of PolyWorld, a neural network that generates precise polygons by extracting object vertices from an image and connecting them optimally by solving a optimal transport problem. The approach extracts local visual descriptors of both edges and vertices from the image, and incorporates global cues and priors by using a Graph Neural Network. In this paper, we proposed an edge-aware attention mechanism as central part of the model, allowing the network to more effectively reason about the scene, and ultimately, to generate more precise polygons. The representation of polygonal scenes has been generalized to enable the application to a wide range of tasks and problem settings. Re:PolyWorld not only outperforms PolyWorld in extracting buildings from aerial images thanks to its combination of vertex and edge analysis, but also reaches state-of-the-art in diverse other tasks, including floorplan reconstruction, wireframe parsing, and architecture reconstruction.

References

[1] Jiacheng Chen, Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2661–2670, 2019. 2

[2] Yuhao Chen, Yifan Wu, Linlin Xu, and Alexander Wong. Quantization in relative gradient angle domain for building polygon estimation. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8360–8367. IEEE, 2021. 2

[3] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013. 5

[4] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the interna-*

tional journal for geographic information and geovisualization, 10(2):112–122, 1973. 6

[5] Nicolas Girard, Dmitriy Smirnov, Justin Solomon, and Yuliya Tarabalka. Regularized building segmentation by frame field learning. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 1805–1808. IEEE, 2020. 2

[6] Nicolas Girard, Dmitriy Smirnov, Justin Solomon, and Yuliya Tarabalka. Polygonal building extraction by frame field learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5891–5900, 2021. 2, 6, 7

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 6

[8] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 626–635, 2018. 7

[9] Yasutaka Furukawa Jiacheng Chen, Yiming Qian. Heat: Holistic edge attention transformer for structured reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 6, 7

[10] Zuoyue Li, Jan Dirk Wegner, and Aurélien Lucchi. Topological map extraction from overhead images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1715–1724, 2019. 2, 6

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6, 7

[12] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018. 6

[13] Sharada Prasanna Mohanty, Jakub Czakon, Kamil A Kaczmarek, Andrzej Pyskir, Piotr Tarasiewicz, Saket Kunwar, Janick Rohrbach, Dave Luo, Manjunath Prasad, Sascha Fleer, et al. Deep learning for understanding satellite imagery: An experimental survey. *Frontiers in Artificial Intelligence*, 3, 2020. 6, 7

[14] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957. 5

[15] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 483–499. Springer, 2016. 2

[16] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 5

[17] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967. 5

[18] Sinisa Stekovic, Mahdi Rad, Friedrich Fraundorfer, and Vincent Lepetit. Montefloor: Extending mcts for reconstructing accurate large-scale floor plans. *arXiv preprint arXiv:2103.11161*, 2021. 2, 6

[19] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 20–36, 2018. 2

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 4

[21] Yifan Xu, Weijian Xu, David Cheung, and Zhuowen Tu. Line segment detection using transformers without edges. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4257–4266, 2021. 6

[22] Nan Xue, Song Bai, Fudong Wang, Gui-Song Xia, Tianfu Wu, and Liangpei Zhang. Learning attraction field representation for robust line segment detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1595–1603, 2019. 2

[23] Nan Xue, Song Bai, Fu-Dong Wang, Gui-Song Xia, Tianfu Wu, Liangpei Zhang, and Philip HS Torr. Learning regional attraction for line segment detection. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1998–2013, 2019. 2

[24] Nan Xue, Tianfu Wu, Song Bai, Fudong Wang, Gui-Song Xia, Liangpei Zhang, and Philip HS Torr. Holistically-attracted wireframe parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2788–2797, 2020. 2, 6, 8

[25] Nan Xue, Tianfu Wu, Song Bai, Fu-Dong Wang, Gui-Song Xia, Liangpei Zhang, and Philip HS Torr. Holistically-attracted wireframe parsing: From supervised to self-supervised learning. *arXiv preprint arXiv:2210.12971*, 2022. 2, 8

[26] Kang Zhao, Jungwon Kang, Jaewook Jung, and Gunho Sohn. Building extraction from satellite images using mask r-cnn with building boundary regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 247–251, 2018. 2

[27] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *European Conference on Computer Vision*, pages 519–535. Springer, 2020. 6, 7

[28] Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):901–914, 2018. 2

[29] Yichao Zhou, Haozhi Qi, and Yi Ma. End-to-end wireframe parsing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 962–971, 2019. 2, 7, 8

[30] Stefano Zorzi, Shabab Bazrafkan, Stefan Habenschuss, and Friedrich Fraundorfer. Polyworld: Polygonal building extraction with graph neural networks in satellite images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1848–1857, 2022. 1, 5, 6, 7

[31] Stefano Zorzi, Ksenia Bittner, and Friedrich Fraundorfer. Machine-learned regularization and polygonization of building segmentation masks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3098–3105. IEEE, 2021. 2

[32] Stefano Zorzi and Friedrich Fraundorfer. Regularization of building boundaries in satellite images using adversarial and regularized losses. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 5140–5143. IEEE, 2019. 2