ISPRS Journal of Photogrammetry and Remote Sensing 170 (2020) 114–126

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

An end-to-end shape modeling framework for vectorized building outline generation from aerial images

Qi Chen [a,1], Lei Wang [b,*,1], Steven L. Waslander [b], Xiuguo Liu [a]

[a] School of Geography and Information Engineering, China University of Geosciences (Wuhan), China
[b] Institute for Aerospace Studies, University of Toronto, Canada

ARTICLE INFO

Keywords:
Building segmentation
Boundary optimization
Automatic mapping
Deep learning
Shape modeling

ABSTRACT

The identification and annotation of buildings has long been a tedious and expensive part of high-precision vector map production. The deep learning techniques such as fully convolution network (FCN) have largely promoted the accuracy of automatic building segmentation from remote sensing images. However, compared with the deep-learning-based building segmentation methods that greatly benefit from data-driven feature learning, the building boundary vector representation generation techniques mainly rely on handcrafted features and high human intervention. These techniques continue to employ manual design and ignore the opportunity of using the rich feature information that can be learned from training data to directly generate vectorized boundary descriptions. Aiming to address this problem, we introduce PolygonCNN, a learnable end-to-end vector shape modeling framework for generating building outlines from aerial images. The framework first performs an FCN-like segmentation to extract initial building contours. Then, by encoding the vertices of the building polygons along with the pooled image features extracted from segmentation step, a modified PointNet is proposed to learn shape priors and predict a polygon vertex deformation to generate refined building vector results. Additionally, we propose 1) a simplify-and-densify sampling strategy to generate homogeneously sampled polygon with well-kept geometric signals for shape prior learning; and 2) a novel loss function for estimating shape similarity between building polygons with vastly different vertex numbers. The experiments on over 10,000 building samples verify that PolygonCNN can generate building vectors with higher vertex-based F1-score than the state-of-the-art method, and simultaneously well maintains the building segmentation accuracy achieved by the FCN-like model.

1. Introduction

An up-to-date high-precision geographic vector map is not only of high significance for applications such as urban planning, change detection, and disaster management, but also plays as an important base map in various location-based business and customer services. Aerial photography is one of the major data sources for high-precision vector map production. In the process of converting aerial images to a vector map, identification and annotation of buildings has long been a tedious and expensive task due to their wide coverage and large quantity in urban areas. Although publicly available vector maps such as OpenStreetMap (Haklay and Weber, 2008) and Ordnance Survey datasets (Hewitt, 2011) can provide mapping information of buildings, the precision of the open-source data is usually limited by problems like incorrect/missing annotations, misalignment errors and over generalization (Vargas-Muñoz et al., 2019; Griffiths and Boehm, 2019).

Automatic building detection from aerial images has been considered an important means to improve the efficiency of vector map production for decades (Paparoditis et al., 1998; Persson et al., 2005; Yang et al., 2018). In recent years, with support of vast training data and sufficient computing capacity, deep learning techniques, such as convolutional neural networks (CNN) (LeCun et al., 1989) and fully convolutional networks (FCN) (Long et al., 2014), have dramatically improved the accuracy of building detection from remote sensing images (Boonpook et al., 2018; Chen et al., 2019; Huang et al., 2019). However, automatic generation of high-quality building vector maps from aerial images is not yet a reality for most built-up areas. This is partly because the deep-learning-based building detection approaches are still facing challenges, such as the low recognition rate for roofs occluded by trees or shadows (Chen et al., 2019) and the relatively

* Corresponding author.
E-mail addresses: chenqi@cug.edu.cn (Q. Chen), lei.wang@robotics.utias.utoronto.ca (L. Wang), stevenw@utias.utoronto.ca (S.L. Waslander), liuxg@cug.edu.cn (X. Liu).
[1] These authors contribute equally to this work.

https://doi.org/10.1016/j.isprsjprs.2020.10.008
Received 9 March 2020; Received in revised form 14 October 2020; Accepted 14 October 2020

poor generalization capability from certain geographic regions to others (Maggiori et al., 2017b). Another issue that may not have received enough attention is that even for the highly accurately detected buildings, slight missed/false detection may still persist along the building boundaries, such that applying polygon simplification techniques to these boundaries could easily generate inaccurate and irregular vector shapes.

Compared with the deep-learning-based building segmentation methods that greatly benefit from data-driven feature learning, the building boundary optimization techniques, especially those designed for making commercially valuable vector results, continue to rely on handcrafted features and high human intervention. Typically, low-level image features (e.g., straight lines or corner points) and a set of manually defined rules (e.g., building angles are likely to be 90 degrees) are often adopted to optimize the initially traced building boundaries (MicroSoft, 2018). These manually designed optimization strategies can generally achieve good simplification and regularization; however, the optimization of building boundaries is usually associated with certain decrease of segmentation accuracy (Zhao et al., 2018; Wei et al., 2019), since the limited generalization capability of handcrafted features can easily cause loss of boundary details for certain types of buildings.

The optimization methods relying on handcrafted features and rules ignore the opportunity of utilizing the rich feature information that can be learned from training data. On the contrary, we argue that a learning-based approach can be developed to model the statistical patterns of the building shape distribution, which we term building *shape priors*. In this study, we propose a learnable end-to-end shape modeling framework, PolygonCNN, for generating building vectors from aerial images. Specifically, PolygonCNN divides the vector generation problem into two successive steps (i.e., building segmentation and shape optimization) and tackles them with different network structures. The building segmentation is performed by an FCN-like structure, initial building shapes (i.e., polygons) are extracted from the segmentation results; the shape optimization is conducted by a one-dimensional (1D) CNN. The 1D CNN, which is a modified version of PointNet (Qi et al., 2017), takes the vertex sequences of the initial polygons as input and optimizes the building shapes by predicting a polygon deformation at the vertex level to generate regularized and map-ready building vector results.

We test PolygonCNN on building instances selected from a large-scale high-resolution aerial imagery dataset. The experimental results verify that the proposed framework outperforms the state-of-the-art method in terms of vertex-based F1-score, thus leading to better vector representation of the identified buildings. Simultaneously, the building segmentation accuracy achieved by the state-of-the-art segmentation network can be well maintained. The main contributions of our work are as follows:

– We propose PolygonCNN, an integrated learning framework that contains an FCN-like structure for building segmentation and a modified PointNet for shape optimization, the whole framework is trained end-to-end.

– We develop a simplify-and-densify strategy for resampling the initially traced building contour, which can generate homogeneously sampled polygon with well-kept geometric signals for shape prior learning.

– By expanding the receptive field of the convolution kernel and applying recursive padding, PointNet is modified to better extract local and global geometric features for building shape optimization.

– We propose a novel loss function for estimating shape similarity between polygons with vastly different vertex numbers.

The remainder of the paper is structured as follows. Section 2 reviews the related work. Section 3 presents the details of the proposed framework. Section 4 introduces the dataset, implementation details and the evaluation metrics. Section 5 shows the evaluation results compared with the state-of-the-art methods and discusses the effectiveness of the design options. Section 6 draws the study conclusions.

2. Related work

2.1. Building segmentation

In general, the existing studies of building segmentation can be categorized into traditional methods and deep-learning-based methods. Traditional methods usually have a bright line between the steps of feature extraction and label classification: the features designed for different operating elements such as pixels (e.g., keypoints or corner points), edges (e.g., straight lines) and/or regions (e.g., texture, context, or shadow evidence) are extracted in advance, the building segments are then generated by applying template matching (Sirmacek and Unsalan, 2009), graph cut (Manno-Kovacs and Ok, 2015), classifiers (e.g., random forest and support vector machine) (Aravena Pelizari et al., 2018; Turker and Koc-San, 2015) or other techniques to the featured elements. Despite many significant achievements, the performance of these methods is largely determined by their handcrafted feature design. Since the complexity and variety of buildings is difficult to be captured and represented by such handcrafted features, these traditional methods usually have limited generalization capability, especially for very-high-resolution aerial images (Wu et al., 2018b).

The deep-learning-based building segmentation methods, mainly enlightened by the theory of CNN, allow for adaptive feature learning from training data. Generally, explicit feature design is not required for these methods, highly discriminative features can be learned from massive labeled training data. The earlier studies usually perform pixel-wise segmentation through a patch-wise classification framework (Guo et al., 2017; Alshehhi et al., 2017), in which each pixel's label is associated with the patch it belongs to. The heavy overlap between patches results in redundant computation and low efficiency; thus, the idea of FCN has become more popular for building segmentation due to its capability of efficiently performing pixel-to-pixel classification (Maggiori et al., 2017a; Bittner et al., 2018). Afterwards, several modified or improved FCN-like models have been proposed and applied to building segmentation. One of the directions of major improvement is utilizing symmetric architectures such as SegNet (Badrinarayanan et al., 2015) and feature pyramid networks (FPN) (Lin et al., 2016) to enhance the spatial information of the final feature map for class prediction (Yang et al., 2018). Additionally, techniques such as multi-scale feature fusion (Zhao et al., 2017; Chen et al., 2019), feature selection (Huang et al., 2019), decision forests (Mi and Chen, 2020) and model ensembles (Marmanis et al., 2018) have been proposed to further improve the segmentation accuracy.

2.2. Building boundary optimization

Segmentation results inevitably have small errors or omissions on building boundaries; therefore, many building boundary optimization methods have been proposed for generating accurate, simple and regular building vector representation from segmented images. A typical scenario that requires boundary optimization is using image features to improve the accuracy of building boundaries extracted from point cloud data (e.g., data generated by airborne laser scanning or dense image matching) (Chen et al., 2016; Dai et al., 2017; Partovi et al., 2017), mostly because images of higher resolution can provide more structural details than the point cloud data. When only image data is available, the target of optimization usually focuses on generating simplified and regularized building boundary shapes. In this case, hand-crafted features or constraint rules such as 90-deg corners or the principal orientation constraint are often applied (Ling et al., 2012; Zhang et al., 2018). However, it is difficult for these low-level features to achieve high generalization performance over diverse buildings that may have distinct boundary shapes. Although these methods would generally produce regularized building shapes, the segmentation accuracy of

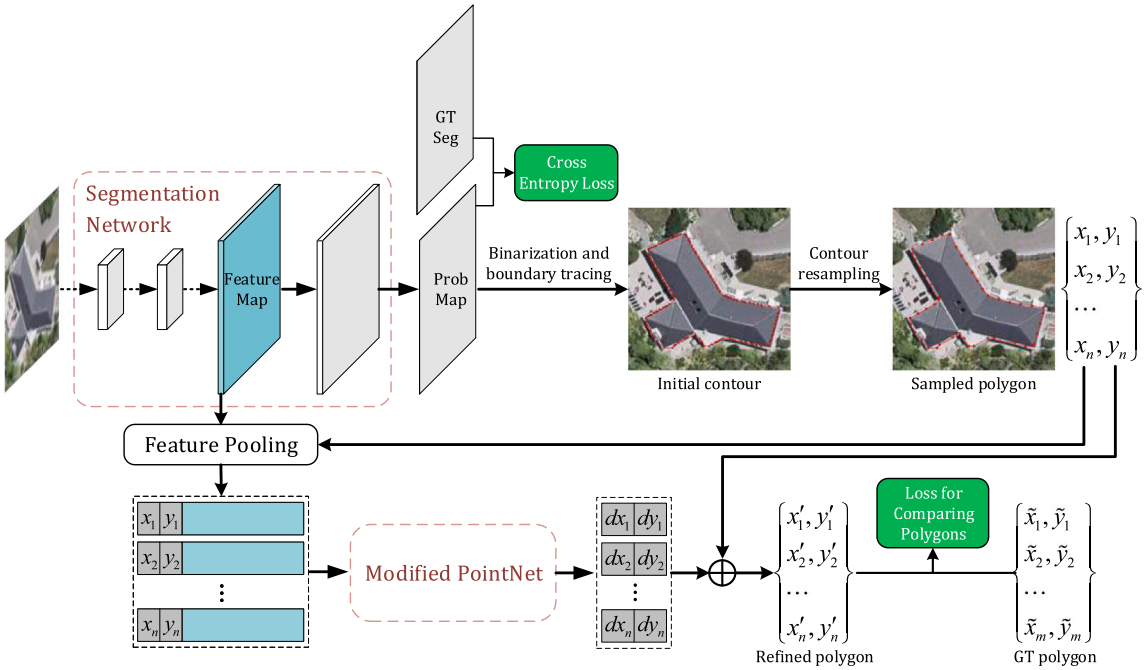

Fig. 1. The architecture of the proposed PolygonCNN framework.

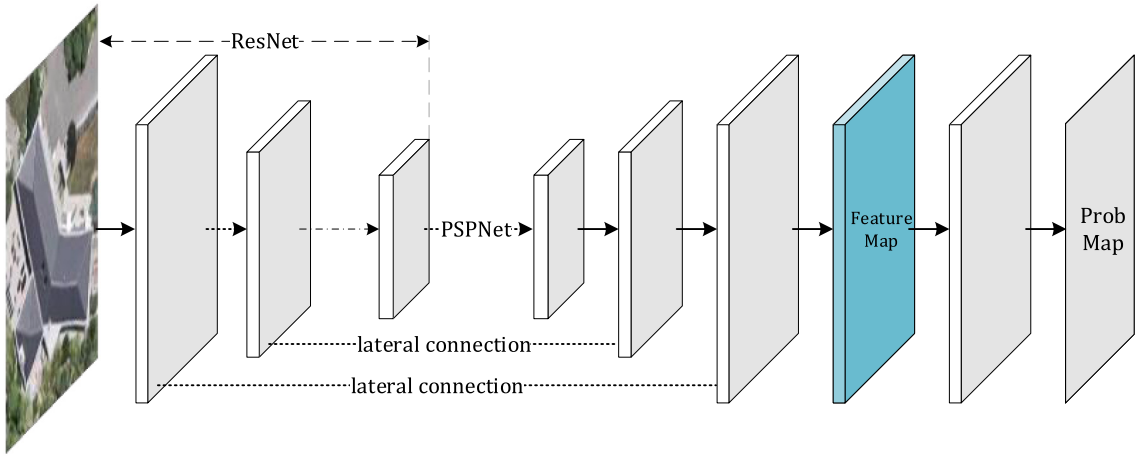

Fig. 2. The structure of the building segmentation network.

the identified buildings tends to decrease during the polygonization process (Zhao et al., 2018; Wei et al., 2019).

Embedding the problem of building boundary extraction/optimization into a deep learning framework is currently a new trend. Lu et al. (2018) directly use building edges as ground truths to train a CNN model that makes boundary predictions; other studies encode building segmentation and boundary extraction as a multi-task learning problem by explicitly adopting boundaries as additional supervision, which evidently makes the two tasks benefit from each other (Marmanis et al., 2018; Volpi and Tuia, 2018; Wu et al., 2018a). However, these approaches stay at modeling building boundary extraction as a pixel classification problem instead of a shape optimization problem; thus, the irregularity of the extracted boundaries is not well solved. Marcos et al. (2018) employ a CNN to learn the parameters of an active contour model for producing building polygons close to the ground truths; Cheng et al. (2019) improve this work by proposing a deep active ray network (DARNet). These studies successfully encode building boundary optimization as a learnable problem, but still fail to take into consideration the simplicity and regularity of building vectors. Another deep-learning-based framework that can explicitly generate object boundaries from images is PolygonRNN (Castrejón et al., 2017); however, its high memory requirement largely limits the spatial resolution of the input images.

A similar work to ours is the recently proposed PolyTransform, which also includes a deforming network for polygon refinement (Liang et al., 2019). However, PolyTransform is not designed specifically for buildings in remote sensing scenes, the simplicity and regularity of the generated vectors is not its concern. Therefore, the two studies actually have a large difference in their design details: (1) we use the 1D CNN (i.e., the modified PointNet) for shape prior learning, which is different from the classic CNN adopted in PolyTransform; (2) we perform a careful vertex resampling for the initial contours before shape optimization, while PolyTransform directly takes the contours as input; and (3) we propose a novel loss function for estimating the similarity between polygons, since the Chamfer Distance loss adopted by PolyTransform performs poorly in our approach (see details in Section 5.2.2).

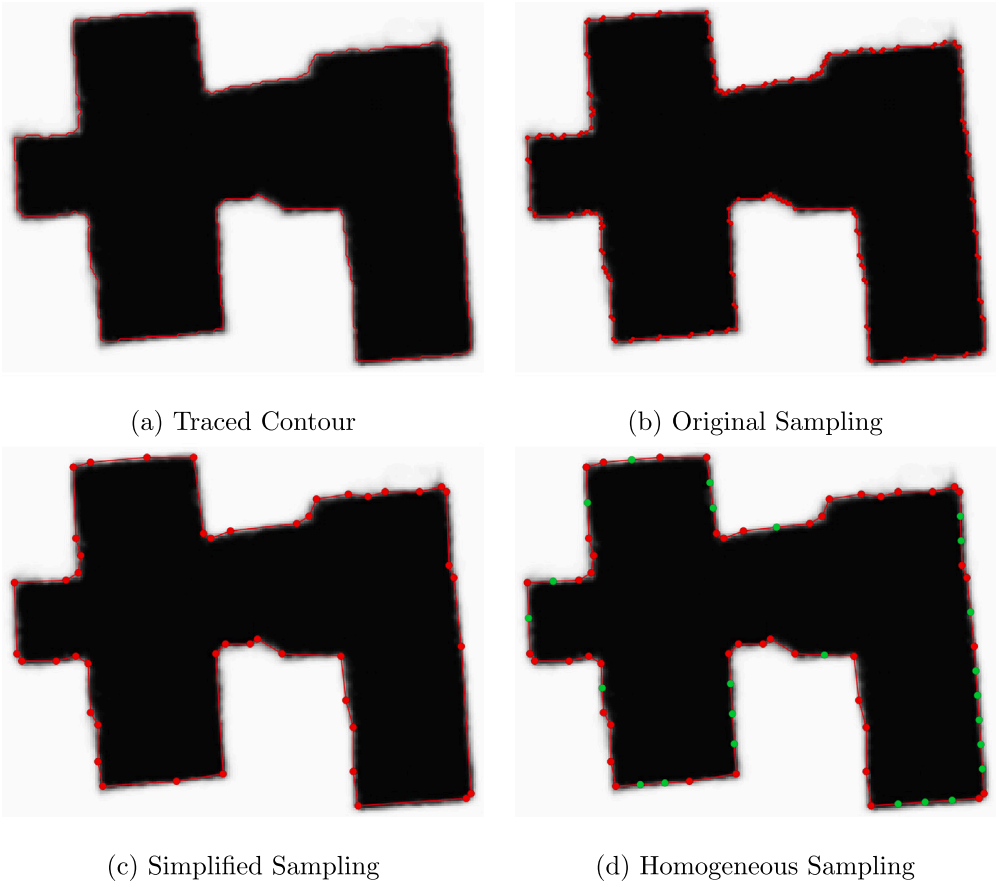

Fig. 3. The traced building contour and polygons with different sampling strategies. In (d), the red dots represent the vertices of the simplified polygons, the green dots represent the additional sampled vertices after applying densification. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

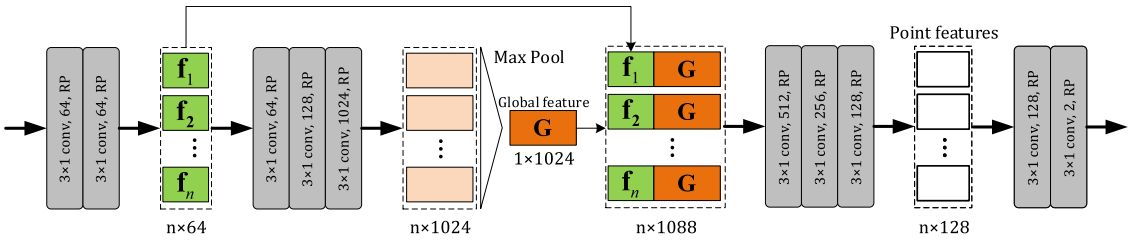

Fig. 4. The structure of the modified PointNet for shape prior Learning. **RP** stands for recursive padding.

2.3. Shape understanding in deep learning

Building boundary optimization can be formulated as a shape deformation problem that takes the sequence of shape vertex coordinates as input, and regresses the offset for each vertex based on shape priors. It is worth noting that there have been several instructive deep learning frameworks designed for learning shape priors of geometric data. PointNet (Qi et al., 2017) and graph convolutional networks such as spectral CNN (Yi et al., 2017) show their capabilities of representing and understanding the geometric features of point clouds, but the learning tasks of these frameworks are more focused on shape classification and segmentation, regression of point locations is currently less researched. Three-dimensional (3D) face reconstruction (Liu et al., 2018) and single-image 3D reconstruction (Wang et al., 2018) are typical shape deformation studies that perform point-wise regression; however, in these approaches, the purpose of the regressor is to predict depth values for a fixed number of points, which is different from the requirements of shape deformation for building boundaries.

Another key factor to construct a feasible deep learning framework for shape deformation is the definition of a loss function. The Chamfer Distance (CD), which sums the projection distance of each point set to the other point set, has been a widely-used metric in recent studies for learning tasks of point cloud data (Fan et al., 2017; Groueix et al., 2018; Sun et al., 2018). CD does not enforce an one-to-one exclusive matching between two point sets, but still requires the predicted shape and targeting shape to be close and have similar number of points. The Earth Mover's distance (EMD), which finds the optimal bi-projection between two sets of points and computes the projection distance, is another metric for point cloud comparison (Fan et al., 2017). The computation of EMD requires the two point sets to have the same number of points. However, in our approach, the vertex numbers of the estimated polygon and the ground-truth polygon can be vastly different; thus, a more proper loss function is needed for comparing polygons that have a large difference in number of vertices.

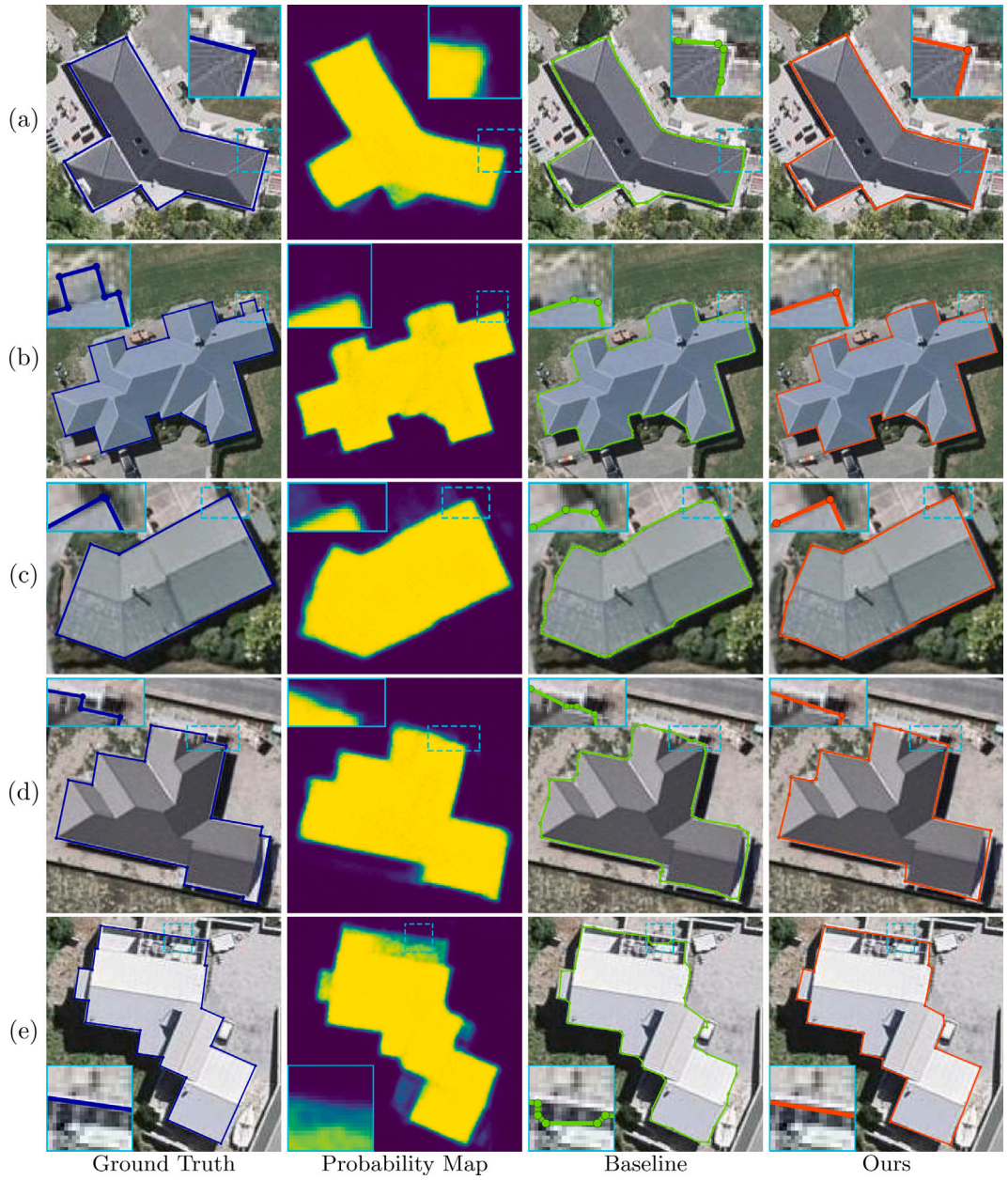

Fig. 5. Examples of generated building vectors for the test set. The ground truths, the vectors generated by the baseline method and our PolygonCNN are in blue, green and red, respectively. The probability maps (where blue means low and yellow means high) are intermediate results generated by the segmentation network. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. Methods

As illustrated in Fig. 1, the proposed PolygonCNN framework mainly consists of a segmentation network and a modified PointNet. The segmentation network is adopted for generating the initial building contour (i.e., polygon), the modified PointNet adjusts the coordinates of the polygon vertices to improve the contour accuracy of the building. Our core idea is to guide the vertices from their initial positions to more accurate or proper positions while preserving the topology of the polygon, so that the quality and regularity of the building shape can be improved. More importantly, after optimizing the positions of vertices, the loss of small geometric details during the polygon simplification process can be reduced, since we do not need to apply a strict threshold to achieve a simplified vector representation. The design details of the framework are introduced below.

3.1. The segmentation network

The pyramid scene parsing network (PSPNet) (Zhao et al., 2017) is now a widely-used FCN-like models for semantic image segmentation. It is also proven very effective in building detection from aerial images (Chen et al., 2019). As shown in Fig. 2, the architecture of PSPNet including ResNet (He et al., 2016) as backbone is used for building segmentation in our approach. Furthermore, considering that the symmetric design of an FPN can further enhance the spatial information, which is helpful for improving the initial contour accuracy of building boundaries, the feature map generated by PSPNet is upsampled twice, the upsampled maps are merged with their corresponding maps from the downsampling pathway by lateral connection (i.e., element-wise addition). The last merged feature map is used to generate the final feature map, which is followed by an inference structure for predicting the probability map.

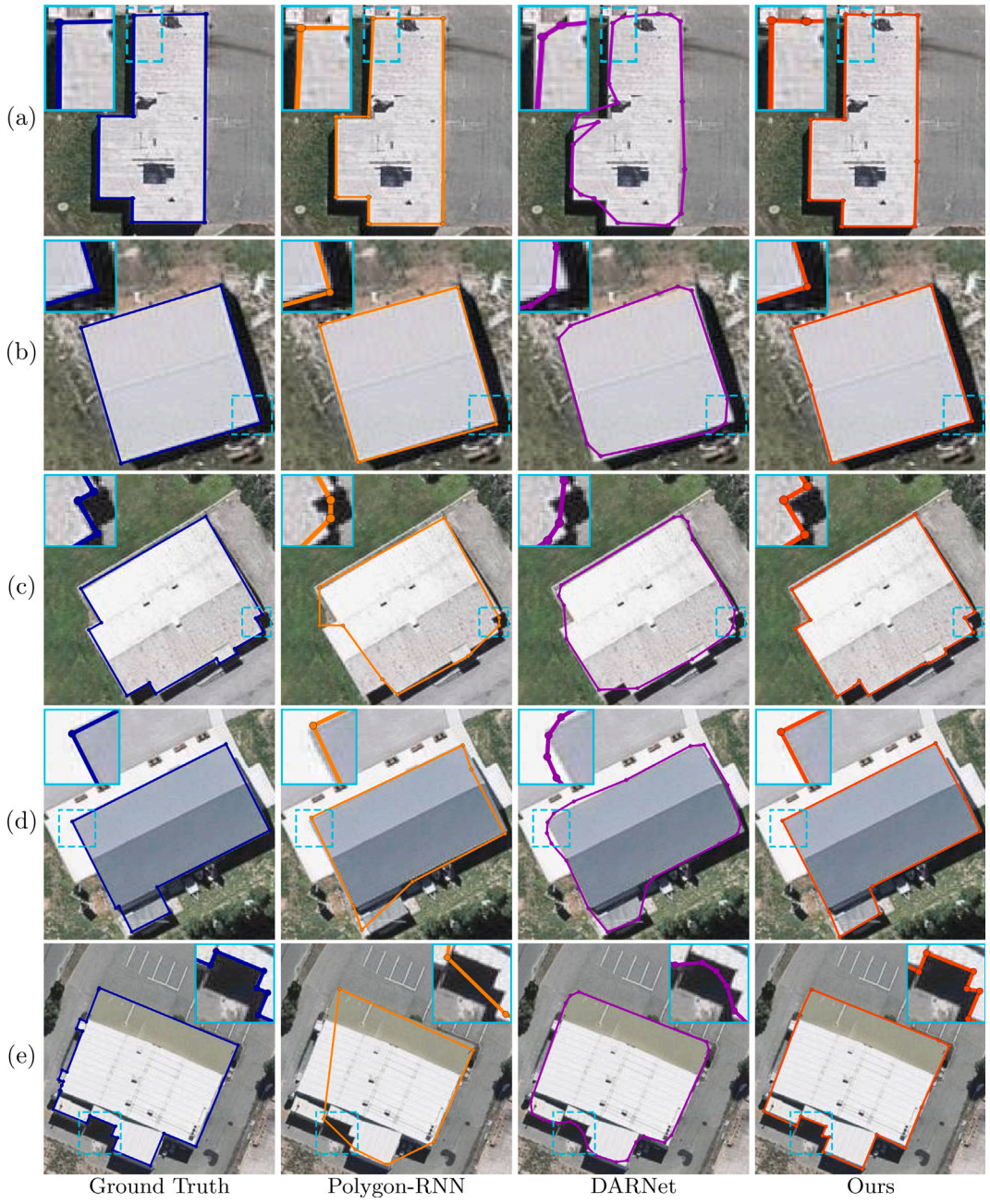

Fig. 6. Examples of building vectors generated by Polygon-RNN, DARNet and the proposed framework. The ground truths, the vectors generated by Polygon-RNN, DARNet and our PolygonCNN are in blue, orange, purple and red, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.2. Contour tracing and resampling

To remove small structures and pixel classification noise, the morphological opening operation with a rectangular structure element is applied on the probability map generated by the segmentation network. The filtered probability map is then binarized and the building boundaries can be extracted by tracing the contours (Suzuki and Be, 1985) of closed areas from the binary map. The traced contour (Fig. 3a) is essentially a sequence of pixels which contains redundant vertices; therefore, it is not an appropriate input for a CNN to learn shape priors of buildings. Removing the intermediate points on straight line segments can reduce the length of sequence without losing any geometric details, but this original sampled polygon (Fig. 3b) still retains far more vertices than are needed for representing the building.

Applying polygon simplification approaches such as the Douglas–Peucker (DP) algorithm (Douglas and Peucker, 1973) can lead to a relatively simplified sampled polygon (Fig. 3c); however, the distance between two consecutive vertices could vary considerably due to the structural changes or slight segmentation errors. In this situation, a fixed-size convolution operation applied to locations of different structure might have very different receptive fields (e.g., a 3×3 convolution kernel could be applied on vertex sequences with significant difference in contour length), which actually brings biases to the features learned from different locations.

Considering this problem, we propose a simplify-and-densify strategy to generate homogeneously sampled polygons as input for the following shape optimization process. First, the initial contour is simplified by the DP algorithm using a relatively loose threshold, the major geometric signals of the contour can therefore be well preserved.

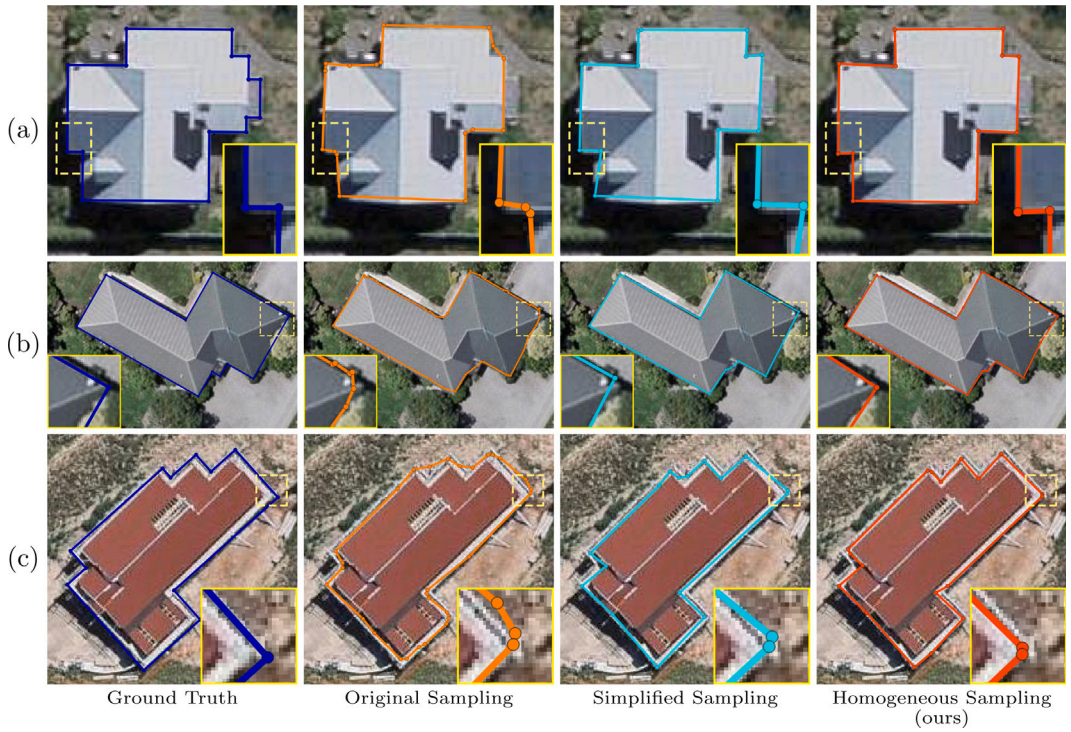

Fig. 7. Examples of building vectors generated by the proposed framework using different polygon sampling methods. The ground truths, the vector results generated based on original sampled polygon, simplified polygon and our homogeneously sampled polygon are in blue, orange, cyan and red, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Then, as Fig. 3d shows, the simplified polygon is densified by adding sampled points along the polygon edges (i.e., sampling additional points evenly on the edges longer than a predefined interval). After this treatment, a sliding convolution kernel can have a similar receptive field throughout the polygon. The experimental results (Section 5.2.1) demonstrate that the proposed strategy performs better than others in terms of vertex-based evaluation metrics.

3.3. The modified pointnet for contour shape optimization

As shown in Fig. 1, after obtaining the homogeneously sampled polygon for the building contour, a feature pooling process, which extracts the feature vector corresponding to every polygon vertex, is applied on the final feature map of the segmentation network. The vertex coordinates and the pooled features are then concatenated and passed to a modified PointNet (i.e., the 1D CNN structure) which takes the combined feature representation as input and optimizes the shape of the building polygon. We choose to use the PointNet structure because it has been demonstrated to have strong performance on classification and segmentation of raw point clouds (Qi et al., 2017); however, to better fit the requirements of shape prior learning for building contours, we propose to make two modifications over the original PointNet:

- **Expanding receptive fields of the convolution layers.** In the original PointNet, the input point sets are unordered, the coordinates of every point are encoded by a 1×1 convolution layer. However, our approach deals with features that have a specific order, and we therefore replace the 1×1 convolution layers with 3×1 convolution layers to learn local context between neighboring vertices of the building polygon.
- **Applying recursive padding.** Zero padding is a generic operation to avoid shrinkage of dimension when conducting convolution at the boundaries of the input feature map with filters larger than 1×1. However, in our approach, applying zero padding to the starting point and end point of the input sequence would ignore the neighborhood relativity between the two points. To overcome this problem, we choose to consider the input point sequence as a closed loop, so that the convolution operation on the point at one end can be performed by "padding" the points at the other end. This strategy, which we term *recursive padding*, ensures that every point in the sequence can contribute its neighboring context to the shape prior learning procedure, and the network can be invariant to the starting point selection of the polygon.

The process flow of the modified PointNet is illustrated in Fig. 4. The first two convolution layers generate local features for every vertex and forms a tensor $[\boldsymbol{f_1}, \boldsymbol{f_2}, \ldots, \boldsymbol{f_n}]$. The following three convolution layers increase the feature dimensions and compute the global feature $\boldsymbol{G}$ through a max pooling function. The global feature vector is concatenated to the local feature vector of each vertex, the combined features are then handled by the following three convolution layers to extract the final per-point features, which are used to predict the deformation vector for every vertex.

3.4. Loss function for estimating similarity between polygons

The refined polygon can be obtained by adding the deformation vector to the input polygon. To make the shape optimization network trainable, a loss function must be defined to estimate the similarity between the refined polygon and the ground-truth polygon. Since the vertex numbers of the refined polygon and ground-truth polygon could be vastly different in our approach, applying metrics such as CD or EMD loss that have been used previously in point cloud comparison may be infeasible or lead to poor performance. Therefore, a novel loss function, $L_{polygon}$, is proposed to compare polygons with different vertex numbers. We define the proposed function as $L_{polygon} = L_{bp} + \lambda L_{rs}$, where L_{bp} and L_{rs} represent two metrics, the *bi-projection loss* and the *relative shape loss*, which penalize the deviation of vertices and line segments, respectively; λ is the weight parameter of L_{rs}.

Fig. 8. Examples of the building vector results optimized by different loss functions. The ground truths, the vector results optimized by CD loss, bi-projection loss and the proposed scheme (bi-projection loss plus relative shape loss) are in blue, yellow, white and red, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.4.1. Bi-projection loss

Considering the difference of the vertex numbers between polygons, we propose to create a one-to-one point correspondence by sampling additional points. For a polygon with more vertices $\boldsymbol{P_m}$ and the other one with fewer vertices $\boldsymbol{P_f}$, we first determine the correspondence for every vertex of $\boldsymbol{P_f}$ by finding its nearest vertex from $\boldsymbol{P_m}$. We name these correspondences as *initial pairs* S_{ip}. Then, for the remaining vertices in $\boldsymbol{P_m}$, the projection point from the vertex to the nearest line segment of $\boldsymbol{P_f}$ is sampled as its corresponding vertex. These correspondences are named as *additional pairs* S_{ap}. After the one-to-one point correspondence is determined, the bi-projection loss can be defined as:

$$L_{bp} = \frac{1}{n}\left(\sum_{(\boldsymbol{p},\bar{\boldsymbol{p}})\in S_{ip}} \|\boldsymbol{p}-\bar{\boldsymbol{p}}\|_2 + \sum_{(\boldsymbol{q},\bar{\boldsymbol{q}})\in S_{ap}} \|\boldsymbol{q}-\bar{\boldsymbol{q}}\|_2\right) \tag{1}$$

where n is the number of total point pairs, $(\boldsymbol{p},\bar{\boldsymbol{p}})$ is a pair of points from the collection of initial pairs S_{ip}, $(\boldsymbol{q},\bar{\boldsymbol{q}})$ is a pair of points from the collection of additional pairs S_{ap}.

3.4.2. Relative shape loss

The bi-projection loss can drive the vertices of the refined polygon to be close to the ground-truth polygon, but for line segments with different lengths, vertex pairs that have the same deviation may lead to varying degrees of shape deformation (e.g., for short line segments, small deviation of the vertex could lead to large shape deformation, which may not be the case for long line segments). Therefore, we propose to use relative shape loss as an additional constraint, which aims at encouraging the matched line segments of the two polygons to be parallel. For each polygon pair, we break the vertex sequence of $\boldsymbol{P_m}$ into continuous line segments and determine their matching segments based on the above one-to-one point correspondence relationship. The relative shape loss can be defined by calculating the sine measure for the angle of the directional vectors of the two matched line segments:

$$L_{rs} = \frac{1}{n}\sum_{(\boldsymbol{l},\bar{\boldsymbol{l}})\in S_{lp}} \frac{\|\boldsymbol{l}\times\bar{\boldsymbol{l}}\|_2}{\|\boldsymbol{l}\|_2\cdot\|\bar{\boldsymbol{l}}\|_2} \tag{2}$$

where n is the number of total line-segment pairs, $\boldsymbol{l}$ and $\bar{\boldsymbol{l}}$ are the 3D directional vectors of the matched line segments with a 0 entry as the third component, S_{lp} indicates the collection of matched line segments.

4. Experimental settings

4.1. Dataset

We evaluate our approach on building instances selected from AIRS (Chen et al., 2019), a large-scale aerial imagery dataset with 7.5 cm resolution and a wide coverage of buildings. Note that since we focus more on the performance of shape modeling for building vectors, following the experimental setup of Polygon-RNN (Castrejón et al., 2017), we assume that a rough bounding box around the building has been given for every sample and randomly select over 10,000 building instances from the AIRS dataset. Considering that the buildings have various sizes, in order to include sufficient background area for each selected building, we expand its context region by 40% of the size of its ground-truth bounding box plus 30 pixels of padding; furthermore, before cropping the aerial image from the expanded box, a random offset (0–30 pixels) is added to the box to avoid giving the model a prior knowledge that the building locates in the center of the image. After this treatment, we separate the samples into two sets, which contain 8131 and 2033 samples for training and testing, respectively.

4.2. Implementation details

We implement and test the proposed PolygonCNN framework in PyTorch (Steiner et al., 2019) on a 64-bit Ubuntu system equipped with an NVIDIA GeForce GTX 1080 Ti GPU. We use ResNet-50 as the backbone of the segmentation network considering the amount of training data. Following its original implementation, the PSPNet architecture of our framework outputs a feature map with 1/8th of the input resolution. After upsampling and feature merging, the final feature map is half-resolution of the input, which is then upsampled to input resolution for inference of segmentation results. For reducing GPU memory consumption, the modified PointNet uses the final feature map with half-resolution during building contour optimization; thus, after refinement, the vertex coordinates of the building polygon are mapped to the input image scale to generate final results of building vectors.

The framework is trained end-to-end with ADAM (Kingma and Ba, 2015) using a learning rate of 0.0001. The batch size is set as 1, so that the size difference of samples does not affect the feasibility of model training. However, for each sample, we still empirically resize excessively small cropped images such that the longer image side has at least 103 pixels to ensure sufficient context information, and also resize excessively large images such that the longer side has at most 553 pixels for GPU memory limitation. Before feeding the vertex coordinates of the homogeneously sampled polygon into the modified PointNet, all the coordinates are normalized to have a mean of 0 and a standard deviation of 1. We set the threshold ε of the DP algorithm as 1 pixel to perform polygon simplification for the traced building contour. The polygon densification process ensures the distance between consecutive vertices is shorter than 10 pixels. Besides 3×1, we also evaluate the convolution kernel sizes of 5×1 and 7×1 for the modified PointNet and observe similar performance. The weight parameter λ in the proposed loss function is set as 1. After shape optimization, the refined polygon is again processed by the DP algorithm with ε of 1 pixel to remove redundant vertices.

4.3. Evaluation metrics

The metric of intersection over union (IoU) (Jaccard, 1912) is used to evaluate the overall performance of building segmentation. Besides that, similar to the estimation of contour accuracy (Perazzi et al., 2016), we compute metrics of F1-score, precision, and recall from a vertex-based perspective to evaluate the vectorization performance of the generated building vectors. Specifically, the refined polygon and the ground-truth polygon are interpreted as two sets of vertices; by setting a buffer for every ground-truth vertex, all the vertices can be classified into true positives, false positives and false negatives, thus the vertex-based F1-score, precision, and recall, which we term *vertex accuracy*, can be computed. Following previous studies (Cheng et al., 2019), for each metric, the average score of the computed values at buffer sizes from 1 to 5 pixel is used for evaluation. We use only vertices instead of the whole contour for evaluation because simplified representation is of high importance in actual mapping production; furthermore, since the post-editing process on generated building vectors is mainly based on vertices, vertex accuracy can better reflect the manual editing work required for the results to be converted to real products.

Table 1
Results on the test set of the building samples. The **IoU** is evaluated on the segmentation results delineated by the finally generated building vectors. The **VertexF**, **VertexP** and **VertexR** represent the metrics of vertex-based F1-score, precision and recall, respectively.

Method	IoU	VertexF	VertexP	VertexR
Segmentation + DP (Baseline)	0.869	0.194	0.137	0.333
PolygonCNN w/o Image feature	0.878	0.316	0.289	0.349
PolygonCNN w/o Joint training	0.878	0.320	0.292	0.354
PolygonCNN (ours)	**0.886**	**0.417**	**0.408**	**0.426**

5. Results and discussion

5.1. Overall results and comparison

Table 1 reports the evaluation results on the test set of the building samples. We use a baseline method which shares the same segmentation network of PolygonCNN for generating traced building contours and simply applies the DP algorithm for polygon simplification. Additionally, we adopt two ablated versions of the proposed framework for comparison. The first one drops the pooled image feature from the input of the modified PointNet but only feeds it with the coordinates; thus, the segmentation network and the shape optimization network work as two separate modules. The second one has the same feed-forward manner as PolygonCNN but trains the two modules separately.

The results show that the baseline method performs poorly in vertex accuracy, especially of the precision; besides that, applying the DP algorithm does decrease the accuracy of building segmentation (from 0.880 to 0.869 in IoU). Even without using the pooled image features, the proposed framework can largely improve the vertex accuracy over the baseline (0.316 vs. 0.194 in F1-score). When the image features are used, the framework without joint training has almost no difference in its performance. In contrast, the end-to-end trained PolygonCNN gains a further improvement on the vertex accuracy (0.417 vs. 0.316 in F1-score); meanwhile, the segmentation accuracy of the framework is also slightly increased, which even outperforms the original accuracy achieved by the segmentation network (0.886 vs. 0.880 in IoU). This indicates that the segmentation task and the shape optimization task of our framework can benefit from each other during the joint training.

Fig. 5 shows examples of building vectors generated by the baseline method and our PolygonCNN framework. The probability maps demonstrate that the main structure of these samples are well detected by the segmentation network; however, slight disturbance still exists, especially at the roof edges and corners. As a consequence, the baseline results show that applying DP with a loose threshold (1 pixel) leaves many redundant vertices, and also some false or missed detection remains. In comparison, our PolygonCNN can use the image feature along with the learned shape priors to refine the vertex positions, which leads to a more accurate vector shape with largely reduced vertices. Note that the DP algorithm applied on the refined polygon uses the same threshold as the baseline. When the building segmentation results have obvious errors or fail to recover certain local details (enlarged

Table 2
Comparison of the evaluation results between DARNet and PolygonCNN. The **IoU** is evaluated on the segmentation results delineated by the finally generated building vectors. The **VertexF**, **VertexP** and **VertexR** represent the metrics of vertex-based F1-score, precision and recall calculated at different buffer size, respectively.

Method	IoU	Vertex accuracy			
		Buffer size	VertexF	VertexP	VertexR
Polygon-RNN	0.677	1 pixel	0.143	0.184	0.117
		2 pixel	0.320	0.412	0.262
		3 pixel	0.417	0.536	0.341
		4 pixel	0.470	0.605	0.385
		5 pixel	0.505	0.650	0.413
		Average	0.371	**0.478**	0.304
DARNet	0.771	1 pixel	0.006	0.005	0.007
		2 pixel	0.021	0.018	0.026
		3 pixel	0.049	0.041	0.061
		4 pixel	0.084	0.070	0.103
		5 pixel	0.134	0.112	0.165
		Average	0.059	0.049	0.073
PolygonCNN	**0.886**	1 pixel	0.137	0.134	0.140
		2 pixel	0.350	0.343	0.357
		3 pixel	0.477	0.467	0.487
		4 pixel	0.543	0.532	0.554
		5 pixel	0.579	0.567	0.591
		Average	**0.417**	0.408	**0.426**

view in Fig. 5b, d), our framework basically cannot correct the existing errors. Instead, the framework guides the initial building shape to be more *regular* from a geometric perspective, which we believe is a good compromise when obtaining perfect results is difficult. More results generated by our framework are shown in Appendix.

We then compare our framework with Polygon-RNN[2] and DARNet (Cheng et al., 2019), which solve the similar problem as ours. For a fair comparison, we train Polygon-RNN and DARNet with the same training set and conduct prediction on the same test set used by PolygonCNN; the generated building boundaries are also processed by the DP algorithm with ε of 1 pixel.

Table 2 reports the evaluation results of the three frameworks. Besides the average score of vertex-based F1-score, precision and recall, the values of these metrics calculated at different buffer sizes are also presented in the table. The results show that our PolygonCNN achieves the highest segmentation accuracy (0.886 in IoU). Polygon-RNN has the worst performance in terms of building segmentation (0.677 in IoU) because it fails to detect complete roof structure for many complicated buildings. The vertex-based precision of Polygon-RNN is the highest among the three, which indicates that this framework performs the best in accurately detecting the vertex points (corners) of the building vectors. Our framework outperforms Polygon-RNN in terms of the overall vertex accuracy (0.417 vs. 0.371 in average F1-score) due to the higher recall rate of the polygon vertices (0.426 vs. 0.304 in average recall). DARNet performs poorly in the three vertex-based metrics at every buffer size, which reflects that this framework can hardly produce polygon vertices that are in 5-pixel buffer area of the ground-truth vertices and generates large number of redundant vertices outside the range. Compared to the original paper, the evaluation values of DARNet decrease sharply, this is largely because they use the whole contour of polygon for metric calculation, while we only use vertices.

The examples of Fig. 6 show that Polygon-RNN can accurately generate the polygon vertices for the buildings with simple structure and plain texture (Fig. 6a, b), but fails to recover complete roof structure when processing more complicated samples (Fig. 6c–e), which could be due to the insufficient capability of its segmentation module. DARNet is basically capable of extracting the main structures of the buildings, but usually fails to accurately capture the small boundary details and sharp corners; thus, redundant vertices can easily be observed from its results (enlarged view in Fig. 6b, d), which is largely responsible for the poor performance in the vertex accuracy estimation. In comparison, our PolygonCNN can obtain better segmentation results for the buildings and also achieve simplified vector representation for different samples.

[2] An unofficial implementation of Polygon-RNN (https://github.com/AlexMa011/pytorch-polygon-rnn) is adopted.

Table 3
The evaluation results of the proposed framework using different polygon sampling strategies. The **IoU** is evaluated on the segmentation results delineated by the finally generated building vectors. The **VertexF**, **VertexP** and **VertexR** represent the metrics of vertex-based F1-score, precision and recall, respectively.

Method	IoU	VertexF	VertexP	VertexR
Original sampling	0.864	0.244	0.197	0.321
Simplified sampling	0.873	0.369	0.359	0.379
Homogeneous sampling (ours)	**0.886**	**0.417**	**0.408**	**0.426**

Table 4
The evaluation results of the proposed framework using different loss functions. The **IoU** is evaluated on the segmentation results delineated by the finally generated building vectors. The **VertexF**, **VertexP** and **VertexR** represent the metrics of vertex-based F1-score, precision and recall, respectively.

Method	IoU	VertexF	VertexP	VertexR
CD loss	0.070	0.038	0.023	0.102
Bi-projection loss	0.849	0.365	0.337	0.397
Bi-projection + Relative Shape loss (ours)	**0.886**	**0.417**	**0.408**	**0.426**

5.2. Analysis of design options

We also evaluate the feasibility of the proposed framework by assessing the performance of alternative design options. We first analyze the impact of different polygon sampling strategies described in Fig. 3, and then compare the results optimized by different loss functions.

5.2.1. Vertex sampling of polygon

Table 3 reports the evaluation results of the proposed framework using three different polygon sampling strategies. For different sampling methods, the refined polygons are all processed by the DP algorithm using the *same* threshold to generate final results. The following comparison shows that although the three methods perform similarly in terms of segmentation accuracy (slight difference in IoU), their performance in vertex accuracy is largely affected by different sampling strategies. The original sampling leads to a much lower accuracy than the other two methods, especially in terms of precision. This is partially because the redundant vertices cannot be removed during the polygon simplification process after shape optimization. Our homogeneous sampling method performs better than the simplified sampling by about 5% in all vertex-based metrics, which quantitatively indicates that the proposed strategy helps the optimization module guide more vertices close to their correct positions.

The examples in Fig. 7 shows that the original sampling could easily result in over-smooth boundaries and inaccurate representation for corners. The simplified sampling can lead to much more simplified vector results; however, the proposed sampling strategy can generally help the framework achieve slightly more accurate vector representation, especially at the corners with less obvious image features (the enlarged view in Fig. 7a, b).

5.2.2. Loss function for comparing polygons

Since CD loss is a widely used function for comparing point cloud, our initial attempt is to test it in the proposed framework. However, as reported in Table 4, the application of CD loss can hardly generate practical results. The bi-projection loss makes it possible to obtain reasonable results and outperforms the baseline in terms of vertex accuracy (0.365 vs. 0.194 in F1-score), largely because this loss function takes

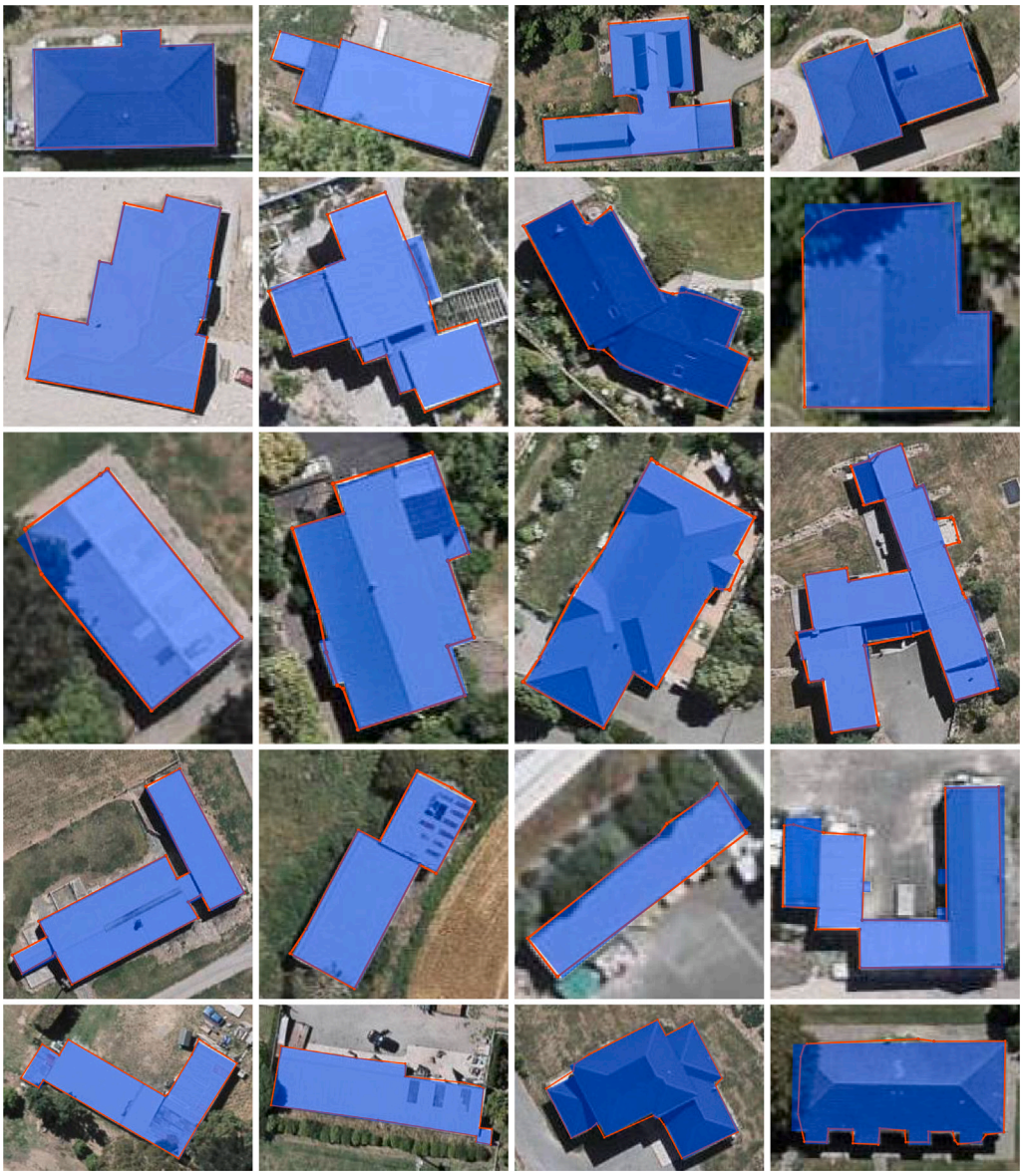

Fig. A.1. More examples of the building vector results generated by our PolygonCNN. The generated vectors are in red; the ground truths are shaded in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

account of the vast difference of the vertex numbers between polygons in our approach. Furthermore, by applying relative shape loss as an additional item, the proposed framework gains more improvement both in segmentation and vertex accuracy. The vertex accuracy improvement is mainly driven by the precision (0.408 vs. 0.337), which indicates that the portion of erroneous vertices among the vector results is effectively reduced.

The examples in Fig. 8 shows that applying CD loss is completely unable to maintain the original shape of the vectors. The difference between applying bi-projection loss only and the proposed scheme mainly lies in the regularity of corners. As shown in the enlarged view of the figures, the proposed loss function is more likely to constrain the corners to have right angles, which to some extent verifies the effectiveness of the angular constraint designed in the relative shape loss.

6. Conclusion

In this work, we propose PolygonCNN, an end-to-end learning framework for generating vectorized building outlines from aerial images. The framework contains a segmentation network for initial building contour extraction and a modified PointNet for shape optimization. After building segmentation and contour initialization, a simplify-and-densify sampling strategy is proposed and applied on the initial contour, which not only generates a homogeneously sampled polygon as input for the modified PointNet but also maintains an accurate representation of the original geometric signals. The modified PointNet refines the vertex coordinates of the polygon by predicting the offset for each vertex. Since the refined polygon and the ground-truth polygon may have a large difference in vertex cardinality, a novel loss function combining bi-projection loss and relative shape loss is proposed to train the framework effectively. The experimental results on a test set with over 2000 building samples demonstrate that our PolygonCNN can generate simplified and regular building vector results and improve

the vertex-based F1-score over the state-of-the-art method. Meanwhile, the feasibility of sampling strategy and loss function adopted by the framework is verified by assessing the performance of alternative design options. A subsequent study will try to use instance segmentation techniques rather than classic semantic segmentation methods to further improve the practicality of the proposed framework. Moreover, experiments on a larger scale are also considered in future works.

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.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (No. 41601506), the Canadian MITACS Elevate program (No. IT15170) and the Fundamental Research Funds for the Central Universities, China (No. CUG190603).

Appendix. More visual results

See Fig. A.1.

References

Alshehhi, R., Marpu, P.R., Woon, W.L., Mura, M.D., 2017. Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks. ISPRS J. Photogramm. Remote Sens. (ISSN: 09242716) 130, 139–149. http://dx.doi.org/10.1016/j.isprsjprs.2017.05.002.

Aravena Pelizari, P., Spröhnle, K., Geiß, C., Schoepfer, E., Plank, S., Taubenböck, H., 2018. Multi-sensor feature fusion for very high spatial resolution built-up area extraction in temporary settlements. Remote Sens. Environ. (ISSN: 00344257) 209 (July 2017), 793–807. http://dx.doi.org/10.1016/j.rse.2018.02.025.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2015. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561, 39 (12) 2481–2495. URL http://mi.eng.cam.ac.uk/projects/segnet/.

Bittner, K., Adam, F., Cui, S., Körner, M., Reinartz, P., 2018. Building footprint extraction from VHR remote sensing images combined with normalized DSMs using fused fully convolutional networks. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. (ISSN: 21511535) 11 (8), 2615–2629. http://dx.doi.org/10.1109/JSTARS.2018.2849363.

Boonpook, W., Tan, Y., Ye, Y., Torteeka, P., Torsri, K., Dong, S., 2018. A deep learning approach on building detection from unmanned aerial vehicle-based images in riverbank monitoring. Sensors (Basel, Switzerland) (ISSN: 14248220) 18 (11), http://dx.doi.org/10.3390/s18113921.

Castrejón, L., Kundu, K., Urtasun, R., Fidler, S., 2017. Annotating object instances with a polygon-RNN. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Vol. 2017-Janua(June). pp. 4485–4493. http://dx.doi.org/10.1109/CVPR.2017.477.

Chen, Q., Wang, S., Liu, X., 2016. An improved snake model for refinement of lidar-derived building roof contours using aerial images. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. (ISSN: 16821750) 41 (July), 583–589. http://dx.doi.org/10.5194/isprsarchives-XLI-B3-583-2016.

Chen, Q., Wang, L., Wu, Y., Wu, G., Guo, Z., Waslander, S.L., 2019. Aerial imagery for roof segmentation: A large-scale dataset towards automatic mapping of buildings. ISPRS J. Photogramm. Remote Sens. (ISSN: 09242716) 147 (October 2018), 42–55. http://dx.doi.org/10.1016/j.isprsjprs.2018.11.011.

Cheng, D., Liao, R., Fidler, S., Urtasun, R., 2019. DARNet: Deep active ray network for building segmentation. URL http://arxiv.org/abs/1905.05889.

Dai, Y., Gong, J., Li, Y., Feng, Q., 2017. Building segmentation and outline extraction from uav image-derived point clouds by a line growing algorithm. Int. J. Digit. Earth (ISSN: 17538955) http://dx.doi.org/10.1080/17538947.2016.1269841.

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. Cartogr.: Int. J. Geogr. Inf. Geovisualization (ISSN: 0317-7173) http://dx.doi.org/10.3138/fm57-6770-u75u-7727.

Fan, H., Su, H., Guibas, L.J., 2017. A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 605–613.

Griffiths, D., Boehm, J., 2019. Improving public data for building segmentation from convolutional neural networks (CNNs) for fused airborne lidar and image data using active contours. ISPRS J. Photogramm. Remote Sens. (ISSN: 09242716) 154 (May), 70–83. http://dx.doi.org/10.1016/j.isprsjprs.2019.05.013, URL https://linkinghub.elsevier.com/retrieve/pii/S0924271619301352.

Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M., 2018. A papier-mâché approach to learning 3d surface generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 216–224.

Guo, Z., Chen, Q., Wu, G., Xu, Y., Shibasaki, R., Shao, X., 2017. Village building identification based on ensemble convolutional neural networks. Sensors (Switzerland) (ISSN: 14248220) 17 (11), 1–22. http://dx.doi.org/10.3390/s17112487.

Haklay, M., Weber, P., 2008. Openstreetmap: User-generated street maps. IEEE Pervasive Comput. (ISSN: 1536-1268) 7 (4), 12–18. http://dx.doi.org/10.1109/MPRV.2008.80.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. http://dx.doi.org/10.1109/CVPR.2016.90, URL http://ieeexplore.ieee.org/document/7780459/ ISSN 1664-1078.

Hewitt, R., 2011. Map of a Nation: A Biography of the Ordnance Survey. Granta Books.

Huang, J., Zhang, X., Xin, Q., Sun, Y., Zhang, P., 2019. Automatic building extraction from high-resolution aerial images and LiDAR data using gated residual refinement network. ISPRS J. Photogramm. Remote Sens. (ISSN: 09242716) 151 (January), 91–105. http://dx.doi.org/10.1016/j.isprsjprs.2019.02.019.

Jaccard, P., 1912. The distribution of the flora in the alphine zone. New Phytol. (ISSN: 1469-8137) http://dx.doi.org/10.1111/j.1469-8137.1912.tb05611.x.

Kingma, D.P., Ba, J.L., 2015. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural Comput. (ISSN: 0899-7667) http://dx.doi.org/10.1162/neco.1989.1.4.541.

Liang, J., Homayounfar, N., Ma, W.-C., Xiong, Y., Hu, R., Urtasun, R., 2019. Polytransform: Deep polygon transformer for instance segmentation. URL http://arxiv.org/abs/1912.02801.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2016. Feature pyramid networks for object detection. http://dx.doi.org/10.1109/CVPR.2017.106, URL http://arxiv.org/abs/1612.03144 ISSN 0006-291X.

Ling, F., Li, X., Xiao, F., Fang, S., Dub, Y., 2012. Object-based sub-pixel mapping of buildings incorporating the prior shape information from remotely sensed imagery. Int. J. Appl. Earth Obs. Geoinf. (ISSN: 15698432) http://dx.doi.org/10.1016/j.jag.2012.02.008.

Liu, F., Zhao, Q., Liu, x., Zeng, D., 2018. Joint face alignment and 3D face reconstruction with application to face recognition. IEEE Trans. Pattern Anal. Mach. Intell. (ISSN: 19393539).

Long, J., Shelhamer, E., Darrell, T., 2014. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. http://dx.doi.org/10.1109/TPAMI.2016.2572683, URL http://arxiv.org/abs/1411.4038 ISSN 01628828.

Lu, T., Ming, D., Lin, X., Hong, Z., Bai, X., Fang, J., 2018. Detecting building edges from high spatial resolution remote sensing imagery using richer convolution features network. Remote Sens. 10 (9), 1496. http://dx.doi.org/10.3390/rs10091496.

Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P., 2017a. Convolutional neural networks for large-scale remote-sensing image classification. IEEE Trans. Geosci. Remote Sens. (ISSN: 0196-2892) 55 (2), 645–657. http://dx.doi.org/10.1109/TGRS.2016.2612821, URL http://ieeexplore.ieee.org/document/7592858/.

Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P., Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P., Semantic, C., 2017b. Can semantic labeling methods generalize to any city ? The inria aerial image labeling benchmark.

Manno-Kovacs, A., Ok, A.O., 2015. Building detection from monocular VHR images by integrated urban area knowledge. IEEE Geosci. Remote Sens. Lett. (ISSN: 1545598X) 12 (10), 2140–2144. http://dx.doi.org/10.1109/LGRS.2015.2452962.

Marcos, D., Tuia, D., Kellenberger, B., Zhang, L., Bai, M., Liao, R., Urtasun, R., 2018. Learning deep structured active contours end-to-end. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 8877–8885. http://dx.doi.org/10.1109/CVPR.2018.00925, URL http://arxiv.org/abs/1803.06329 ISSN 0636919.

Marmanis, D., Schindler, K., Wegner, J.D., Galliani, S., Datcu, M., Stilla, U., 2018. Classification with an edge: Improving semantic image segmentation with boundary detection. ISPRS J. Photogramm. Remote Sens. (ISSN: 09242716) 135, 158–172. http://dx.doi.org/10.1016/j.isprsjprs.2017.11.009.

Mi, L., Chen, Z., 2020. Superpixel-enhanced deep neural forest for remote sensing image semantic segmentation. ISPRS J. Photogramm. Remote Sens. (ISSN: 09242716) 159 (November 2019), 140–152. http://dx.doi.org/10.1016/j.isprsjprs.2019.11.006.

MicroSoft, 2018. Computer generated building footprints for the United States. Website https://github.com/Microsoft/USBuildingFootprints.

Paparoditis, N., Cord, M., Jordan, M., Cocquerez, J.P., 1998. Building detection and reconstruction from mid- and high-resolution aerial imagery. Comput. Vis. Image Underst. (ISSN: 10773142) 72 (2), 122–142. http://dx.doi.org/10.1006/cviu.1998.0722.

Partovi, T., Bahmanyar, R., Kraus, T., Reinartz, P., 2017. Building outline extraction using a heuristic approach based on generalization of line segments. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. (ISSN: 21511535) http://dx.doi.org/10.1109/JSTARS.2016.2611861.

Perazzi, F., Pont-Tuset, J., McWilliams, B., Gool, L.V., Gross, M., Sorkine-Hornung, A., 2016. A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. ISBN: 9781467388504, http://dx.doi.org/10.1109/CVPR.2016.85.

Persson, M., Sandvall, M., Duckett, T., 2005. Automatic building detection from aerial images for mobile robot mapping. In: Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA. pp. 273–278. http://dx.doi.org/10.1109/cira.2005.1554289.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. ISBN: 9781538604571, http://dx.doi.org/10.1109/CVPR.2017.16.

Sirmacek, B., Unsalan, C., 2009. Urban-area and building detection using SIFT keypoints and graph theory. IEEE Trans. Geosci. Remote Sens. (ISSN: 0196-2892) 47 (4), 1156–1167. http://dx.doi.org/10.1109/TGRS.2008.2008440.

Steiner, B., Devito, Z., Chintala, S., Gross, S., Paszke, A., Massa, F., Lerer, A., Chanan, G., Lin, Z., Yang, E., Desmaison, A., Tejani, A., Kopf, A., Bradbury, J., Antiga, L., Raison, M., Gimelshein, N., Chilamkurthy, S., Killeen, T., Fang, L., Bai, J., 2019. Pytorch: An imperative style, high-performance deep learning library. NeuroIPS (NeurIPS).

Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T., 2018. Pix3d: Dataset and methods for single-image 3d shape modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2974–2983.

Suzuki, S., Be, K.A., 1985. Topological structural analysis of digitized binary images by border following. Comput. Vis. Graph. Image Process. (ISSN: 0734189X) http://dx.doi.org/10.1016/0734-189X(85)90016-7.

Turker, M., Koc-San, D., 2015. Building extraction from high-resolution optical spaceborne images using the integration of support vector machine (SVM) classification, hough transformation and perceptual grouping. Int. J. Appl. Earth Obs. Geoinf. (ISSN: 1872826X) http://dx.doi.org/10.1016/j.jag.2014.06.016.

Vargas-Muñoz, J.E., Lobry, S., Falcão, A.X., Tuia, D., 2019. Correcting rural building annotations in openstreetmap using convolutional neural networks. ISPRS J. Photogramm. Remote Sens. (ISSN: 09242716) 147 (May 2018), 283–293. http://dx.doi.org/10.1016/j.isprsjprs.2018.11.010.

Volpi, M., Tuia, D., 2018. Deep multi-task learning for a geographically-regularized semantic segmentation of aerial images. ISPRS J. Photogramm. Remote Sens. (ISSN: 09242716) 144 (June), 48–60. http://dx.doi.org/10.1016/j.isprsjprs.2018.06.007, URL https://linkinghub.elsevier.com/retrieve/pii/S0924271618301692.

Wang, N., Zhang, Y., Li, Z., 2018. Pixel2mesh - generating meshes from single RGB images. Eccv.

Wei, S., Ji, S., Lu, M., 2019. Toward automatic building footprint delineation from aerial images using CNN and regularization. IEEE Trans. Geosci. Remote Sens. (ISSN: 0196-2892) PP, 1–12. http://dx.doi.org/10.1109/tgrs.2019.2954461.

Wu, G., Guo, Z., Shi, X., Chen, Q., Xu, Y., Shibasaki, R., Shao, X., 2018a. A boundary regulated network for accurate roof segmentation and outline extraction. Remote Sens. (ISSN: 20724292) 10 (8), 1–19. http://dx.doi.org/10.3390/rs10081195.

Wu, G., Shao, X., Guo, Z., Chen, Q., Yuan, W., Shi, X., Xu, Y., Shibasaki, R., 2018b. Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks. Remote Sens. (ISSN: 2072-4292) http://dx.doi.org/10.3390/rs10030407.

Yang, H.L., Yuan, J., Lunga, D., Laverdiere, M., Rose, A., Bhaduri, B., 2018. Building extraction at scale using convolutional neural network: Mapping of the United States. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. (ISSN: 21511535) 11 (8), 2600–2614. http://dx.doi.org/10.1109/JSTARS.2018.2835377.

Yi, L., Su, H., Guo, X., Guibas, L., 2017. SyncSpecCNN: Synchronized spectral CNN for 3D shape segmentation. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Vol. 2017-Janua. ISBN: 9781538604571, pp. 6584–6592. http://dx.doi.org/10.1109/CVPR.2017.697.

Zhang, C., Hu, Y., Cui, W., 2018. Semiautomatic right-angle building extraction from very high-resolution aerial images using graph cuts with star shape constraint and regularization. J. Appl. Remote Sens. http://dx.doi.org/10.1117/1.jrs.12.026005.

Zhao, K., Kang, J., Jung, J., Sohn, G., Street, K., Drive, M., York, N., Mb, O.N., 2018. Building extraction from satellite images using mask R-CNN with building boundary regularization. In: CVPR Workshops. pp. 247–251, URL https://www.topcoder.com/spacenet.

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., Limited, S.G., 2017. Pyramid scene parsing network. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. ISBN: 9781538604571, http://dx.doi.org/10.1109/CVPR.2017.660.