




A Building Shape Vectorization Hierarchy From VHR Remote Sensing Imagery Combined DCNNs-Based Edge Detection and PCA-Based Corner Extraction

Xiang Wen , Xing Li, Wenquan Han, Erzhu Li, Wei Liu , Lianpeng Zhang , Yihu Zhu, Shengli Wang, and Sibao Hao

Abstract—The automatic vectorization of building shape from very high resolution remote sensing imagery is fundamental in many fields, such as urban management and geodatabase updating. Recently, deep convolutional neural networks (DCNNs) have been successfully used for building edge detection, but the results are raster images rather than vectorized maps and do not meet the requirements of many applications. Although there are some algorithms for converting raster images into vector maps, such vector maps often have too many vector points and irregular shapes. This article proposed a building shape vectorization hierarchy, which combined DCNNs-based building edge detection and a corner extraction algorithm based on principle component analysis for rapidly extracting building corners from the building edges. Experiments on the Jiangbei New Area Buildings and Massachusetts Buildings datasets showed that compared with the state-of-the-art corner detectors, the building vector corners extracted using our proposed algorithm had fewer breakpoints and isolated points, and our building vector boundaries were more complete and regular. In addition, the building shapes extracted using our hierarchy were 7.94% higher than the nonmaximum suppression method in terms of relaxed overall accuracy on the Massachusetts dataset. Overall, our proposed hierarchy is effective for building shape vectorization.

Index Terms—Building shape vectorization, corner extraction, deep convolutional neural network (DCNN), edge detection, very high resolution (VHR) remote sensing imagery.

I. INTRODUCTION

AS THE most important feature in urban environment, obtaining accurate building vector shapes is of great significance in many fields, such as cartography and urban planning [1]. However, the vast variations in building appearances indicate that building extraction still relies heavily on manual work [2]. To improve the efficiency of building extraction, it is necessary to study the automatization of building vector shape extraction from very high resolution (VHR) remote sensing images.

Commonly, the building boundaries are extracted to generate a building vector map, so the building edge detection is the premise of building vector shape extraction. Among the building edge detection methods, deep convolutional neural networks (DCNNs)-based edge detection networks, such as HED [3], RCF [4], CED [5], BDCN [6], Dexined [7], and DRC [8] have the most development potential, and have achieved the state-of-the-art performances on the BSDS500 [9] dataset. We conducted relevant experiments, and proved that the DCNNs-based edge detection networks are feasible for building edge extraction from VHR remote sensing images. ME-Net [10] achieved the highest accuracy in detecting the clearest and crispest building edges. The detection results of the building edges are shown in columns 3–7 of Fig. 1. Although the DCNNs-based edge detection networks showed intuitive building edge results on the large-area Massachusetts Buildings dataset in Fig. 1, we can clearly see that the results of building edge probability maps are not perfect as the building edges are thick and they cannot be vectorized directly. Hence, we aimed to vectorize crisper building edges and delineate more accurate building boundaries based on these building edge probability maps.

Considering the problem that the building edge extracted by DCNNs is too thick, the nonmaximum suppression (NMS) is the most common method to sharpen and vectorize the building edge probability map. Although this method can reduce the width of the edge to one pixel, the final building shape is irregular and there are many noise points. Ming et al. [11]

Manuscript received 29 January 2022; revised 23 November 2022; accepted 17 December 2022. Date of publication 22 December 2022; date of current version 29 December 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 41801327, in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant KYCX20_2365, in part by the Jiangsu Province Land and Resources Science and Technology Plan Project under Grant 2021046, and in part by the Key Laboratory of Land Satellite Remote Sensing Application, Ministry of Natural Resources of the People's Republic of China under Grant KLSMNR-K202206. (Corresponding author: Lianpeng Zhang.)

Xiang Wen is with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou 221116, China, and also with the School of Geographic and Oceanographic Sciences, Nanjing University, Nanjing 210023, China (e-mail: wenxiang@jsnu.edu.cn).

Xing Li, Erzhu Li, Wei Liu, and Lianpeng Zhang are with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou 221116, China (e-mail: lixing@jsnu.edu.cn; liezrs2018@jsnu.edu.cn; liuw@jsnu.edu.cn; zhanglp@jsnu.edu.cn).

Wenquan Han is with the Nanjing Institute of Surveying, Mapping & Geotechnical Investigation, Co., Ltd., Nanjing 210019, China (e-mail: hanwq@njcky.com).

Yihu Zhu, Shengli Wang, and Sibao Hao are with the Jiangsu Geologic Surveying and Mapping Institute, Nanjing 211102, China (e-mail: zhuyihoo@jsdzch.com; wsl586@jsdzch.com; hsb@jsdzch.com).

Digital Object Identifier 10.1109/JSTARS.2022.3231348

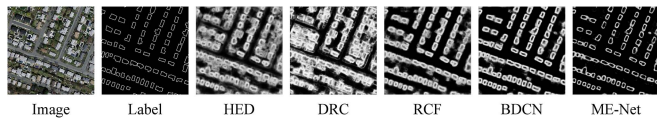


Fig. 1. Sample of the Massachusetts Buildings dataset: The first column shows the original image, the second column shows the building edges label, and columns 3–7 show the building edge probability maps of HED, DRC, RCF, BDCN, and ME-Net, respectively.

described a geomorphological concept for extracting accurate building edge points, and generating building edges based on the edge point map for refining the edge probability map. However, some broken and incomplete building edges remain after their algorithm processing, and further article is necessary to optimize building vector shape extraction.

Generally, the building vector shape extraction belongs to the field of image vector extraction in computer vision. The image vectorization methods can be roughly divided into four categories: Hough-transformation-based method [12], thinning-based method [13], contour-based method [14], and corner-based method [15], [16]. In this article, the geometric features of buildings are more obvious in VHR remote sensing imagery. Among all the building geometric features, the corner feature is one of the most direct evidence of buildings because it can effectively describe the shape and structure of buildings with the least amount of data. In recent years, there have been some successful corner detectors, such as Moravec [17], Harris [18], SUSAN [19], FAST [20], [21], CPDA [22], and CSS [23]. These methods extract corners by calculating the changes of the gradient amplitude, the gradient direction in the nearby area, and the curvature of contour. However, we found that these corner extraction operators are not accurate enough to extract building corners. To extract the precise building corners and produce the building shapes, Xia et al. [24] used the anisotropic-scale junctions detector [25] and proposed a geometric building index for detecting both the exact position and accurate building shape. In addition, Li et al. [26] designed an end-to-end learnable deep-learning architecture, PolyMapper, which extracts keypoints and edge evidence of building footprints, and combines each building in each tile for producing the building vector map. As an extension, a bottom-up instance segmentation method was proposed [27] that extracts building corners by detecting all peaks from the predicted heatmap of building boundaries, and then groups the corners into building boundaries in a simple geometric way. However, this method relies heavily on the extraction results of DCNNs and cannot correct the irregular predictions of building boundaries, so the building shape results need to be further regularized.

The main purpose of this article is to design a complete hierarchy, which contains several continuous and standardized steps for automatically extracting the building vector shape from VHR remote sensing imagery. In our hierarchy, motivated by the idea of corner extraction methods above, we proposed a principle component analysis (PCA)-based corner extraction algorithm for extracting building corners from the thick building edges predicted using DCNNs, and grouped these building corners into more regular and accurate building vector boundaries. In summary, the primary contributions of this article are as follows:

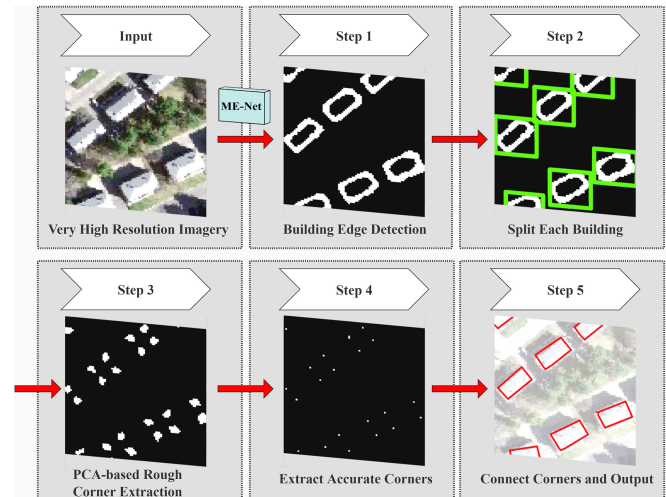


Fig. 2. Main flow chart of our proposed building shape vectorization hierarchy.

- 1) We constructed a hierarchy for extracting building vector shapes, involving five steps: the first involved the building edge detection, the second involved splitting each independent building, the third involved searching the building rough corners using PCA-based rough corner extraction, the fourth involved extracting accurate corners by mean-shift clustering algorithm, and the last involved connecting corners and vectorizing them. This hierarchy can automatically produce the high-precision building vector map from the VHR remote sensing imagery, and the main flow chart of hierarchy is illustrated in Fig. 2.
- 2) This article proposed a PCA-based corner extraction algorithm for extracting building corners from the building edge image of DCNNs-based edge detection network. In a broad sense, this algorithm consists of the second to fifth steps of our hierarchy, and it can quickly search and extract the building corners based on the principal component direction of building edges.
- 3) We compared the building corner results of our hierarchy with the state-of-the-art corner detectors. Meanwhile, we also compared the building vector shape results with the NMS method on the Jiangbei New Area Buildings and Massachusetts Buildings datasets, and both the results demonstrated the universality of our proposed hierarchy for extracting building vector shapes.

The rest of this article is organized as follows. In Section II, the overall architecture of our proposed hierarchy is introduced in detail. Section III illustrates the experiments and comparison results of building vector shapes among different extraction methods. Discussion and conclusions regarding this article are presented in Sections IV and V, respectively.

II. METHODOLOGY

This article presents a hierarchy for the building shape vectorization from VHR remote sensing imagery. This hierarchy applies two core modules: the first is the edge detection network for automatically detecting building edges, and we selected the ME-Net as the practical network due to its relatively crisp and

accurate building edge result (input: VHR remote sensing image and output: building edge image). The other module is our proposed PCA-based corner extraction algorithm for automatic extraction and connection of building corners based on the building edges (input: building edge image and output: building vector map).

A. Overall Hierarchy

In this article, our proposed hierarchy can be divided into the following five consecutive steps in detail:

- 1) For the remote sensing imagery, the building edges are automatically detected using the state-of-the-art DCNNs-based edge detection network, ME-Net.
- 2) Each independent building is divided from the building edge result of ME-Net according to the minimum circumscribed rectangle.
- 3) The rough corners of each building are searched according to the first principle component direction of the building edges and three supplementary angles.
- 4) The accurate corners are extracted by finding the center of rough corners using the mean-shift clustering algorithm.
- 5) The building corners are connected in turn when they have complementary angles in the stored directions in the third step, then the result is vectorized as the final building vector map.

Our proposed building shape vectorization hierarchy was implemented using Anaconda 4.8.2 and Python 3.6.10, and the pseudocode of the hierarchy is described in Algorithm 1. In addition, the experimental details of the first four steps are described in Sections II-B–E.

B. ME-Net

The first step in our proposed hierarchy is the application of DCNNs-based edge detection network for the building edges. The network selection is very significant because the crisper building edges will directly improve the performance of the building corner extraction and shape vectorization in later steps. We noticed that ME-Net [10] produced relatively crisp and accurate building edges. In Section IV-A, we have demonstrated that ME-Net is more suitable in our hierarchy than the other state-of-the-art edge detection networks.

The ME-Net is a multiscale DCNN for extracting building edges from VHR remote sensing images. This network consists of five side layers, and each side layer has the same important modules: the convolution module, the max pooling module, the scale enhancement module, the upsampling block, and the erosion module (EM). This network follows the transmission of bidirectional cascade structure [6] for supervising the edge feature information of five side layers with different specific scales, and the transmission is shown in the red dotted line in Fig. 3. The EM is significant as it refines the building edges of the feature map in each scale and improves the performance of the network.

We implemented the ME-Net using PyTorch1.4.0 and Pytorchvision0.5.0, and set the training parameters of the initial learning rate to 1e-6, momentum to 0.9, and batch size to 1,

Algorithm 1: Our Building Shape Vectorization Hierarchy.

Input: VHR remote sensing imagery I .

Output: building corner map O_1 , building boundary map O_2 .

/ Step 1: building edge detection **//**

1: Initialize all the parameters of the DCNNs-based edge detection network, ME-Net, and feed I into the network.

2: Compute $P \leftarrow$ building edge probability map generated by one forward pass of the ME-Net.

3: Set the classification threshold to 0.5 for all the pixels in P , and store the edge pixels in B .

4: **return** $B \leftarrow$ building edge binary map.

/ Step 2: split each independent building **//**

5: Initialize L_1, L_x, L_y with empty lists.

6: Compute $C_1 \leftarrow$ the list of contours of all the buildings by using `skimage.measure.find_contours(B, 0.1)`.

7: **for all** the contours of each building $C_2 \in C_1$ **do**

8: **for all** each contour pixel $p \in C_2$ **do**

9: $x, y \leftarrow$ coordinates of p .

10: $L_x.add(int(x+0.5))$.

11: $L_y.add(int(y+0.5))$.

12: **end for**

13: $L_1.add([\max(L_x), \min(L_x), \max(L_y), \min(L_y)])$.

14: **end for**

15: **return** $L_1 \leftarrow$ the list of minimum circumscribed rectangular for each independent building in B .

/ Step 3: PCA-based rough corner extraction **//**

16: Compute $\theta_{pca} \leftarrow$ four search directions of rough corners in L_1 by using

math.atan(($C_2.pca.components[1]$)/($C_2.pca.components[0]$)) and $+90^\circ, +180^\circ, +270^\circ$.

17: **for all** each pixel $p \in L_1$ **do**

18: **for all** $\theta \in \theta_{pca}$ **do**

19: $N_1 \leftarrow$ the number of non-edge pixels along θ in B .

20: **if** $N_1 \leq 2$ **then**

21: $T.add(p, \theta)$.

22: **end if**

23: **end for**

24: $N_2 \leftarrow$ the number of search direction θ for p in B .

25: **if** $N_2 < 2$ **then**

26: $T.remove(p, \theta)$.

27: **end if**

28: **end for**

29: **return** $T \leftarrow$ the rough corner pixels with the search direction.

/ Step 4: extract accurate corners **//**

30: Set $MS \leftarrow$ a clustering function by using **sklearn.cluster**.

MeanShift(bandwidth = 4, bin_seeding = True).

31: **for all** each rough corner pixel $p \in T$ **do**

32: $MS.fix(p)$.

33: $O_1.add(MS.cluster_centers)$.

34: **end for**

35: **return** $O_1 \leftarrow$ building corner map.

/ Step 5: connect corners **//**


```

36: for all each accurate corner pixel  $p, q \in O_1$  do
37:  $D_1 \leftarrow$  distance between  $p$  and  $q$  in  $B$ .
38:  $D_2 \leftarrow$  distance between  $p$  and  $q$  in  $O_1$ .
39:  $D_{theta} \leftarrow$  difference in search directions of  $p$  and  $q$ .
40: if  $D_2 \leq D_1$  and  $|D_{theta} - 180| \leq 20$  then
41:  $p_i \leftarrow$  the points from  $p$  to  $q$ .
42:  $O_2.add(p_i)$ .
43: end if
44: end for
45: return  $O_2 \leftarrow$  building boundary map.

```

which are same as the parameters of the published study [10]. Since the ME-Net produced the building edge probability map with values between 0 and 1, we set a threshold [28], [29], [30] for classifying both the edge and nonedge pixels, and the threshold is generally 0.5 [31]. In this article, based on the edge pixels, we can extract the building corners and in later steps of our hierarchy vectorize the building shapes.

C. Splitting Each Independent Building

To extract the building corners rapidly, the second step of our proposed hierarchy is designed to split each independent building by distinguishing the minimum bounding rectangle of all the buildings. We first applied a find-contours function [32] in Python-Skimage, which can detect the iso-valued contours in a two-dimensional array for a given level value, and produce the (x, y) coordinates along the contour. Then, we extract the edge contour of each building in the building edge binary map using the find-contours function, and effectively distinguish all the buildings in turn. In this article, we set the level value of finding contours to 0.1. Fig. 4(b) shows the result of the edge contour map, and the edge contour of each building is marked using different colors.

Furthermore, we counted the coordinate values of each building in the edge contour map, and defined the minimum coordinate value as the top left vertex of the minimum circumscribed rectangle; the maximum coordinate value as the lower right vertex of the minimum circumscribed rectangle and the range of minimum circumscribed rectangle of each building is shown in Fig. 4(c). After performing this step, we only need to search for building corners within each green rectangle, this boosts the efficiency of the whole hierarchy.

D. PCA-Based Rough Corner Extraction

The number of building corners is very small in the original image, thus, to extract all the corners without omission, the third step of our proposed hierarchy is designed to search for building rough corners using our PCA-based corner extraction algorithm.

Depending on the range of the minimum circumscribed rectangle of each building after the second step in Section II-C, our corner extraction algorithm mainly extracts corners from the building edges along four directions (the first principle component direction and its three supplementary angles derived by rotating 90° , 180° , and 270° horizontally), the reason for this

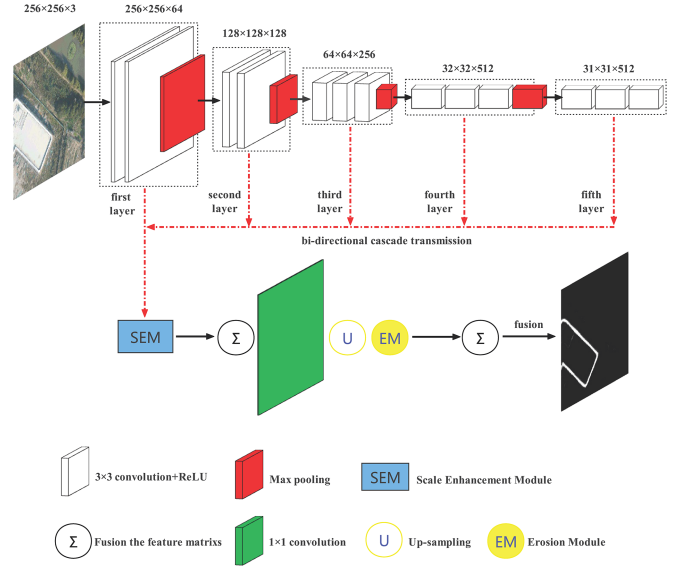


Fig. 3. Architecture of the ME-Net for building edge detection.

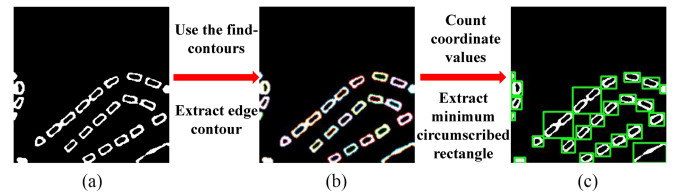


Fig. 4. Details of the second step of our proposed hierarchy. (a) Building edge binary map of ME-Net. (b) Edge contour map with the (x, y) coordinates after using the find-contours function. (c) Minimum circumscribed rectangle result for each building.

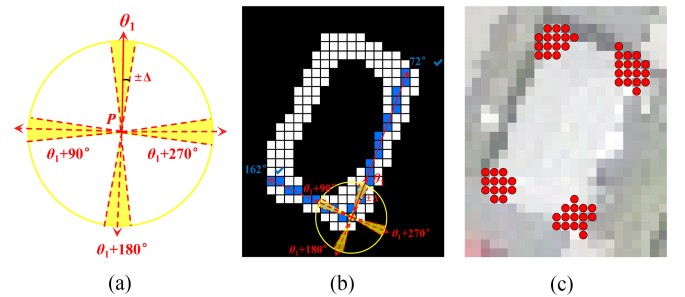


Fig. 5. Example of the third step of our proposed hierarchy: (a) search directions of our PCA-based corner extraction algorithm; (b) search details for one pixel in the building edges; and (c) result of building rough corners.

is that we find that the first principle component direction of building edges (called θ_1) is approximately equal to the actual direction of building. Fig. 5(a) shows the search directions of our algorithm for a point P , considering that there is a small error between θ_1 and the actual building direction, we add the angles less than Δ (set to 10°) in the range of the yellow sector to θ_1 .

In order to obtain these directions effectively, we added a function of PCA [33] in Python-Sklearn, which uses full singular value decomposition (SVD) or a randomized truncated SVD [34] to obtain the principle component directions of the edge

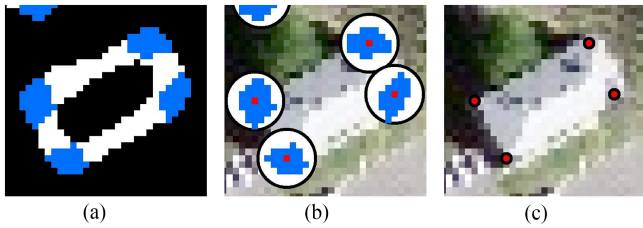


Fig. 6. Example of the fourth step of our proposed hierarchy: (a) building rough corner map; (b) details of clustering the rough corners with the bandwidth of 4; and (c) result of the building corner map.

pixels of each building. The full SVD and randomized truncated SVD are given as follows:

$$A = U \Sigma V^T \quad (1)$$

$$A_{m \times n} = U_{m \times l} \Sigma_{l \times l} V_{n \times l}^T$$

$$A \approx B C \quad (2)$$

$$A_{m \times n} \approx B_{m \times k} C_{k \times n}$$

where A , U , V , B , and C are the real matrices, and Σ is a rectangular, diagonal matrix composed of nonnegative and diagonal elements arranged in descending order. The inner dimension k is the specified dimension of the decomposition matrices of B and C , and (2) is often used to express a low-rank factorization of the given matrix A .

Then, our PCA-based corner extraction algorithm searched for rough corners, pixel by pixel, within the range of each building. As shown in Fig. 5(b), when we judge whether the yellow pixel is a building corner pixel, first, we take this point as the starting point and search the building edge pixel (white pixel) along four directions (θ_1 , $\theta_1 + 90^\circ$, $\theta_1 + 180^\circ$, and $\theta_1 + 270^\circ$), and allow less than two nonedge pixels (black pixel) in one direction. The four search directions can replace searching all directions of 360° , and greatly improve the execution speed of our algorithm. Then, as shown in the two extended red lines, because there are no nonedge pixels along the search directions of 72° and 162° , the two directions are stored as the final directions of the yellow pixel. Eventually, if the yellow pixel has the vertical angles in the stored directions, we mark it as the building rough corner pixel and vectorize it as the rough corner point, and the result of all the building rough corners is shown in Fig. 5(c).

E. Extracting Accurate Corners by Mean-Shift Clustering

Although the third step prevents missing of the building corners, as shown in Fig. 6(a), the building rough corner map has a problem that the corners are gathered in a certain range. Therefore, the fourth step of our proposed hierarchy is designed to extract accurate corners from the building rough corner map.

The details of extracting accurate corners are shown in Fig. 6(b), we set the bandwidth to 4 to cluster all the rough corners within the range of 4 pixels (white circles), and use the mean-shift [35] to train the coordinates of the rough corners. This clustering algorithm is used to find the centroid within these dense building rough corners, and to update the centroid candidate to the average coordinate value of the corner samples. Notably, the center points (red pixels) after clustering the rough

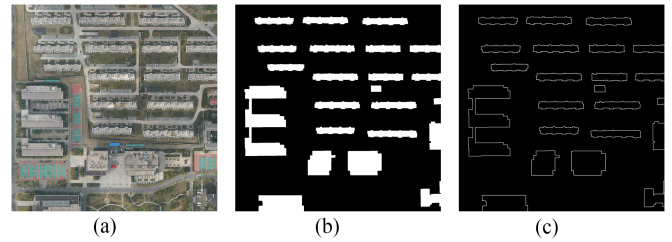


Fig. 7. Large example with a size of 1280×1280 pixels on the Jiangbei New Area Buildings dataset: (a) original image; (b) building regions label; and (c) building edges label.

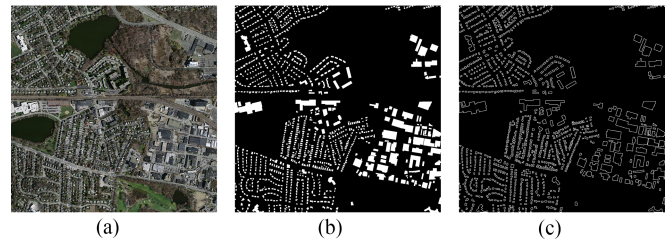


Fig. 8. Massachusetts Buildings dataset sample: (a) original image; (b) original building regions label; and (c) building edges label.

corners not only have the exact point positions, but also have more accurate search angles than the repeated and redundant search angles of rough corners. These center points are vectorized as they provide more accurate corners in the building corner map in Fig. 6(c).

III. EXPERIMENTS AND RESULTS

In this section, we describe the testing of extraction results of the building corners, building boundaries, and building regions of our proposed hierarchy on the Jiangbei New Area Buildings and Massachusetts Buildings datasets, and systematically compare the evaluation metrics and visualization results with other state-of-the-art methods.

A. Datasets

1) *Jiangbei New Area*: The Jiangbei New Area Buildings dataset presented by Wen et al. [10] covers 53.67 km^2 of the Jiangbei New Area, Nanjing City, Jiangsu Province, China. As shown in Fig. 7, this dataset includes aerial images from unmanned aerial vehicles with ground resolution of 0.3 m.

As suggested by Wen et al. [10], to avoid the issue of overfitting the convolution neural networks, the images and label maps were clipped and augmented through rotation. The augmentation of training set is used to extract more complex and diverse buildings. Then, the training, validation, and test sets consist of 8000, 100, and 106 patches with a size of 256×256 pixels.

2) *Massachusetts*: The Massachusetts Buildings dataset [36] is an aerial image building dataset of the Boston Area, and this dataset covers 364.5 km^2 in 151 images with a size of 1500×1500 pixels each. To meet the requirements of building

edge detection, as is shown in Fig. 8(c), Wen et al. [10] and Lu et al. [11] supplied the building edges label maps of this dataset.

Similar to the Jiangbei New Area Buildings dataset, the Massachusetts Buildings dataset is split into a training set (10 500 patches), a validation set (100 patches), and a test set (250 patches) with no overlapping areas. Each patch has a size of 256×256 pixels and a ground resolution of 1 m.

B. Evaluation Metrics

In order to assess the ability to extract building shapes, we used five conventional metrics in this article: overall accuracy (OA), precision, recall, F1-score (F1), and intersection over union (IoU)

$$OA = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

$$F_1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

$$IoU = \frac{TP}{TP + FP + FN} \quad (7)$$

where TP, FP, TN, and FN represent the true positive, false positive, true negative, and false negative predictions, respectively.

The ground truth of manual marked building label will inevitably produce errors, thus, we used relaxed precision [36] and relaxed recall for achieving a fairer evaluation result. The relaxed precision is defined as the fraction of predicted building pixels that are within ρ pixels of a true building pixel, whereas relaxed recall is defined as the fraction of true building pixels within the ρ pixels of a predicted building pixel [37]. In this article, the parameter ρ is set to 3 [38] for all the quantitative experiments.

C. Building Corner Comparison Results With Corner Extraction Methods

Our proposed hierarchy for building shape vectorization is based on the corner extraction method. To prove the effectiveness of our hierarchy, we compared our building corner results with several well-known corner detectors: Harris [18], SUSAN [19], FAST [20], [21], and CPDA [22]. In addition, we also made a comprehensive comparison with the state-of-the-art corner detection model ECFRNet [39] based on deep learning. The visual comparison results for the public Massachusetts Buildings dataset are shown in Fig. 9. It is evident that the corner detectors of Harris [18], SUSAN [19], and FAST [20], [21] produced incorrect building corner maps. In addition, the deep-learning model of ECFRNet [39] and the multiscale corner detector of CPDA [22] were superior and had clearer building corners, but still missed the corners of some small buildings on the dataset.

For the buildings with different shapes and sizes in the first five rows of Fig. 9, the ground truth of building corners in the



Fig. 9. Some building samples of the Massachusetts Buildings dataset and the building corner extraction results of different corner detectors: (a) original image; (b) ground truth; the building corners of (c) Harris; (d) SUSAN; (e) FAST; (f) CPDA; (g) ECFRNet; and (h) ours.

Massachusetts Buildings dataset have large deviation and these errors will greatly affect the accuracy of building corners because there are few true corner samples. However, our proposed hierarchy still extracted the most complete and accurate building corners. For the adjacent buildings in the sixth to eighth row: the Harris, SUSAN, and FAST corner detectors misclassified many points on the building ridge, whereas our proposed hierarchy distinguished the corners of each building effectively. In the 9th to 10th row, under some extreme conditions, such as haze and thin cloud, the corner detection ability of our method for the fuzzy images was not affected, while the other methods detected fewer corners because the contrast of pixels was reduced. Furthermore, our hierarchy showed outstanding performance in the vectorization of building boundary and building region based on the extracted corners, and the performance of our hierarchy is compared with the NMS method in Section III-D.

D. Building Shape Comparison Results With the NMS Method

Normally, the NMS method is used to refine the building edge results of DCNNs and to produce the building vector shape. In this article, we proposed the PCA-based corner extraction algorithm instead of NMS method, in our building shape vectorization hierarchy. To evaluate the advantages of our proposed algorithm, we compared our building shape results

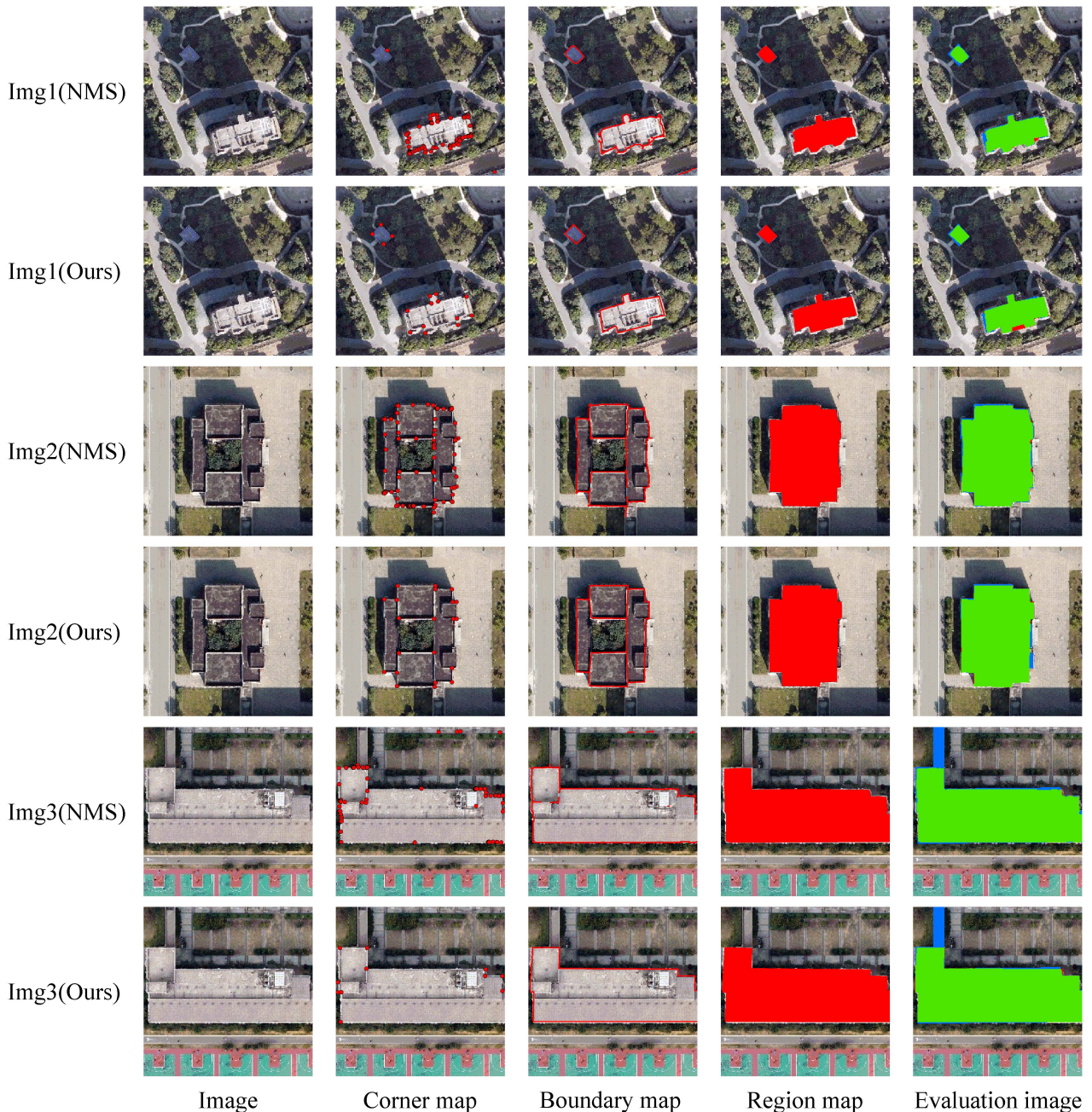


Fig. 10. Three examples of the Jiangbei New Area Buildings test set. Columns 1–5 are original images, building extraction results of corner maps, boundary maps, region maps, and evaluation images, respectively. The green, red, blue, and background pixels in the evaluation images represent true positive, false positive, false negative, and true negative predictions, respectively.

and quantitative metrics with the results obtained with the NMS method on the two datasets.

1) *Results from the Jiangbei New Area Buildings Dataset:* Fig. 10 shows three typical images of the Jiangbei New Area Buildings test set with a size of 256×256 pixels. The odd rows are the original images, corner maps, boundary maps, region maps, and evaluation images produced by the NMS method, and the even rows are those from our algorithm. We produced the evaluation images by evaluating the building regions with the ground truth labels. It is worth mentioning that since the NMS

method cannot directly extract the building corners, the corner map is generated as the endpoint of the boundary map.

As shown in the first image, when extracting the vector shapes of angular residential buildings, although our algorithm misclassified a small area of the roof with more false positives (red pixels in the evaluation image), we detected building corners and building boundaries more accurately than the NMS method. For the building covered with trees in the second image, the boundary extracted by the NMS method is partially broken, whereas our algorithm better judged the building corners and

TABLE I

EVALUATION RESULTS COMPARING THE NMS METHOD AND OUR ALGORITHM ON THREE IMAGES FROM THE JIANGBEI NEW AREA BUILDINGS TEST SET, AND THE BEST VALUES OF COLUMNS 3–5 ARE MARKED AS BOLD

Images	Method	OA (%)	F1 (%)	IoU (%)
Image 1	NMS	99.35	94.60	89.76
	Ours	99.35	94.87	90.24
Image 2	NMS	99.19	98.19	96.45
	Ours	99.31	98.49	97.02
Image 3	NMS	97.06	95.50	91.38
	Ours	97.44	96.13	92.55
Mean	NMS	98.53	96.10	92.53
	Ours	98.70	96.50	93.27

provided more regular building shapes. When extracting the school building shapes in the third image, both methods missed a small passage. However, the building boundaries extracted by our algorithm are tidier. In conclusion, our algorithm performed better and generated more regular building vector shapes for the complex architectures in the Jiangbei New Area Buildings dataset.

Our algorithm not only extracts more regular building shapes, but also has better performance in the metrics. Table I shows the relaxed OA, F1, and IoU values of the three images generated using the NMS method and our building shape extraction algorithm. Note that our algorithm outperforms the NMS method for all three metrics. For the three images, our algorithm in relaxed OA, F1-score, and IoU on average increased by 0.17%, 0.40%, and 0.74%, respectively.

Moreover, our algorithm has achieved the best performance for the test images of the Jiangbei New Area Buildings dataset, and the mean OA even reached 98.70%. However, since this building dataset is built by ourselves, these test images may not be representative enough. To assess the transferability of our algorithm, we also compared the performances using the public Massachusetts Buildings dataset, and the results will be discussed in the second part of Section III-D.

2) *Results from the Massachusetts Buildings Dataset:* Similar to Fig. 10, Fig. 11 presents three representative test images of the Massachusetts Buildings dataset. The buildings of this dataset are densely distributed; thus, the region map results of each building for both methods are not complete enough. The NMS method misses more building shapes than our algorithm with more false negative predictions (blue pixels) shown in the evaluation image.

In addition, as shown in the large building in the lower left area of the sixth image, our algorithm can produce a better building corner map with more exact corner positions. Furthermore, our boundary maps are square and more in line with the geometric characteristics of real buildings. In summary, our proposed algorithm is superior to the NMS method based on the extraction results of building vector maps from left to right.

Table II lists the relaxed quantitative evaluation metrics of the two methods on the Massachusetts Buildings test set. Compared

TABLE II

EVALUATION RESULTS OF THREE REPRESENTATIVE IMAGES AND THE WHOLE MASSACHUSETTS BUILDINGS TEST SET, AND THE BEST VALUES ARE MARKED AS BOLD

Images	Method	OA (%)	F1 (%)	IoU (%)
Image 4	NMS	92.17	70.08	53.94
	Ours	94.76	83.05	71.01
Image 5	NMS	95.23	66.19	49.47
	Ours	96.20	76.80	62.33
Image 6	NMS	94.13	74.17	58.94
	Ours	96.37	86.88	76.80
Test set	NMS	79.46	52.22	37.22
	Ours	87.40	80.26	66.55

with the Jiangbei New Area Buildings dataset, the three metrics dropped sharply because of the lower resolution of the Massachusetts dataset. As indicated in the Table II, our algorithm exceeds the NMS method by 7.94%, 28.04%, and 29.33% for the whole test set in relaxed OA, F1, and IoU, respectively. One reason is that the NMS method often extracts broken building shapes and misses some smaller buildings, so it has poor performance for extracting building vector shapes based on the input building edge maps by DCNNs on the Massachusetts Buildings dataset. The other reason is that our algorithm can extract the accurate building corners and produce a correct building shape, so our algorithm achieved the relaxed OA of 87.40% on the test set, and even obtained 96.37% on the sixth test image of this dataset.

As mentioned in the introduction of Section I, the input building edge maps of DCNNs have a problem interpreting thick edges, and some adjacent buildings were connected by mistake. To analyze the performance of our algorithm on the Massachusetts Buildings dataset further, we filtered out these wrongly connected buildings, and we segmented 3249 complete and independent buildings in the Massachusetts Buildings test set. The relaxed OA, F1, and IoU curves of all the buildings are shown in Fig. 12. The building shape result of the NMS method has broken building boundary and incomplete building region issues, so the NMS method achieves a lower IoU than even equals 0 for some buildings.

In addition, the OA of our algorithm exceeded 90.00% on average for each building as shown by the red curve in Fig. 12. Moreover, almost all the curves of our algorithm are clearly higher than those of the NMS, indicating that our algorithm has a greater improvement in the Massachusetts Buildings dataset. In summary, our PCA-based corner extraction algorithm and our hierarchy had an improved capability for more accurate building shape vectorization.

IV. DISCUSSION

A. Ablation Analysis

In this article, our building shape vectorization hierarchy contains two modules: building edge detection and our

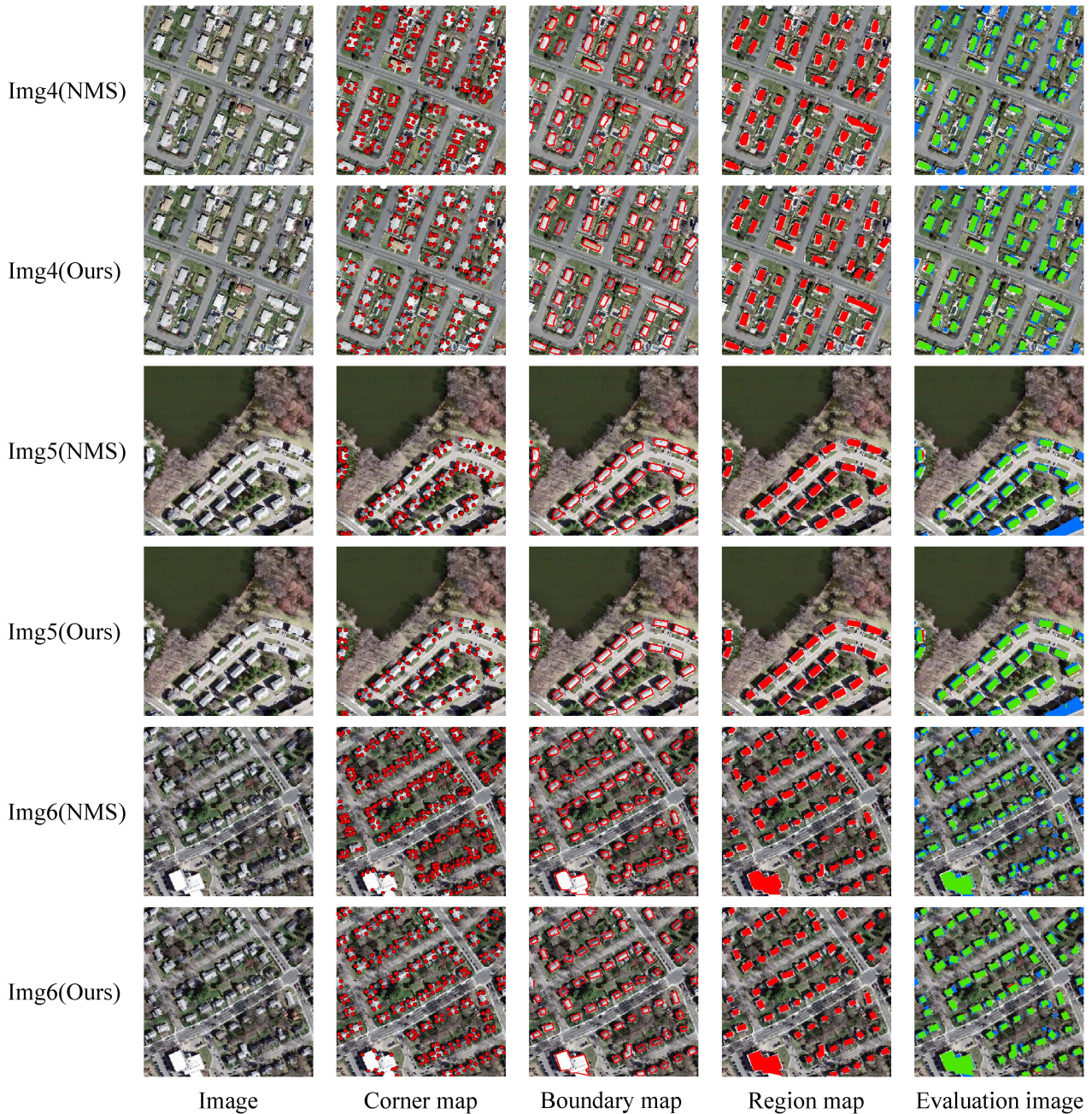


Fig. 11. Three examples of the Massachusetts Buildings test set with a size of 256×256 pixels. The columns 1–5 are original images, building extraction results of corner maps, boundary maps, region maps, and evaluation images using the NMS method or our algorithm, respectively.

proposed PCA-based corner extraction algorithm, and we applied the ME-Net for the building edge detection. To demonstrate the improvement of the ME-Net and our corner extraction algorithm, we performed the ablation analysis for these two modules in our hierarchy. More specifically, we combined five edge detection models (HED [3], RCF [4], BDCN [6], DRC [8], and ME-Net [10]) with two-edge-based building vector shape extraction method (the NMS method and our corner extraction algorithm), and conducted the comparative building shape extraction experiments on the Massachusetts Buildings test set.

Table III lists the relaxed quantitative results of the building shape results using different combinations on the Massachusetts Buildings test set. We also compared the run-times of different models, and the results are shown in the last column of Table III. For every aerial image, the average time cost of models using our proposed PCA-based corner extraction algorithm is 1.08 s, which is faster than 1.25 s using the NMS method. In addition, when extracting building shapes based on the five edge detection models of DRC, HED, RCF, BDCN, and ME-Net, all the metrics of the NMS method scored lower than those of our algorithm. For the building shape extraction results of the NMS method,

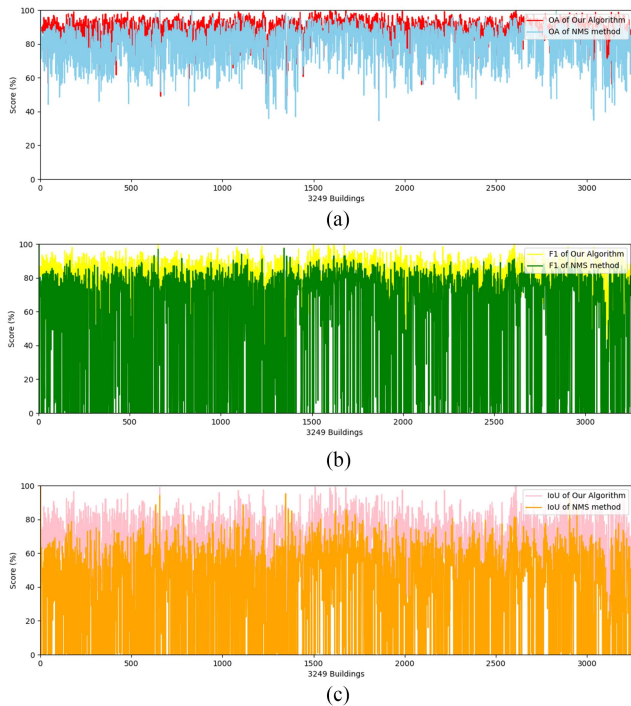


Fig. 12. OA, F1, and IoU of all the 3249 independent buildings on the Massachusetts Buildings test set. The red, yellow, and pink curves represent the OA, F1, and IoU of our algorithm, whereas the blue, green, and orange curves represent those metrics for the NMS method.

TABLE III
QUANTITATIVE RESULTS AND TIME COST OF COMBINATIONS GENERATED BY FIVE EDGE DETECTION MODELS AND TWO BUILDING VECTOR SHAPE EXTRACTION METHODS ON THE MASSACHUSETTS BUILDINGS TEST SET

Model	Method	OA (%)	F1 (%)	IoU (%)	Time (s)
DRC	+ NMS	72.50	21.70	13.65	1.34
	+ Ours	81.38	65.87	48.27	1.33
HED	+ NMS	75.97	40.19	25.27	1.19
	+ Ours	83.44	72.44	54.09	0.99
RCF	+ NMS	79.06	52.90	35.63	1.19
	+ Ours	83.85	73.59	54.34	0.86
BDCN	+ NMS	80.90	57.43	41.96	1.22
	+ Ours	86.68	79.18	63.71	0.91
ME-Net	+ NMS	79.46	52.22	37.22	1.29
	+ Ours	87.40	80.26	66.55	1.32

the combination of the BDCN and the NMS method surpassed 1.44%, 5.21%, and 4.74% than the combination of ME-Net and the NMS method in terms of OA, F1, and IoU, respectively. Furthermore, of all 1 combinations, the architecture of ME-Net and our algorithm achieved the best performance in the three metrics of 87.40%, 80.26%, and 66.55%. This proves that the ME-Net and our corner extraction algorithm play a vital role in improving the performance of our building shape vectorization hierarchy.

B. Regularization and Correction of Building Edges

To better manage our algorithm, there are three kinds of parameters in this article. First, in the Step 1 of Algorithm 1, the classification threshold is used for transforming the building edge probability map into a building edge binary map. Since various building types in different datasets need to be measured uniformly, this threshold value is generally 0.5. Second, in the Step 2 of Algorithm 1, the parameter of find_contours level is used for producing the (x, y) coordinates along the building contour. We carried out relevant experiments and found that under the premise of the operation efficiency of algorithm, the value of 0.1 represents the minimum difference between the edge coordinates, so that the most accurate range of each building can be detected. Third, in the Step 4 of Algorithm 1, the parameter of bandwidth level is used for clustering rough corners in a certain range. Considering the resolution of the dataset and the width of building edge, we tried to change this parameter from 1 to 10, and the value of 4 obviously extracted the most accurate corners.

Some buildings are covered by shadow in remote sensing images, and from these images DCNNs-based edge detection networks will produce broken building edges. The NMS method is unable to make up the broken edges and produces fractured building boundaries and incomplete building regions. However, our hierarchy can better deal with such buildings for two reasons: one is that we relaxed the limit (allowing two nonedge pixels) for searching corners in our proposed PCA-based corner extraction algorithm, to improve corner extraction from the broken building edges. The second reason is that we construct the building vector map based on the extracted corners rather than based on scattered points, so we can produce more regular building boundaries and regions.

Although our hierarchy has an improved capacity for regularizing broken and distorted building edges in the Jiangbei New Area Buildings and the Massachusetts Buildings datasets, we mainly focused on correcting the most common quadrilateral buildings for a high-speed performance. Hence, when our hierarchy is applied for extracting the unconventional building vector shapes such as triangle or circle, it is necessary to add the additional directions of searching corners in the third step of our hierarchy.

V. CONCLUSION

In this article, we used the state-of-the-art DCNNs-based edge detection network, ME-Net, for producing the building edge image, but it cannot be further vectorized due to the thick edges. Although the NMS method can meet the purpose of edge thinning, there is still the problem of irregular building boundaries and incomplete building regions.

To extract the regular building vector shapes from VHR remote sensing imagery automatically, this article constructed a new hierarchy that consisted of five continuous steps. The first is to detect building edges using the DCNN of ME-Net from the VHR remote sensing imagery. In a broad sense, the second to fifth steps are designed as a PCA-based corner extraction algorithm. Specifically, the second step is to split each independent building by producing the minimum circumscribed

rectangle; the third is to search building rough corners through PCA-based rough corner extraction (this step analyzes the actual direction of each building and searches building corners along the principle component direction of building edges instead of searching all the directions); the fourth is to extract accurate corners based on the mean-shift clustering algorithm; and the final step is to connect all the qualified corners in turn, and then produce the building vector map. To verify the effectiveness of our vectorization hierarchy fully, this article conducted comparative experiments on the Jiangbei New Area Buildings and Massachusetts Buildings datasets. The extraction results of building vector shapes by our hierarchy are more accurate and regular and we also achieved the best performance in terms of the quantitative evaluation metrics.

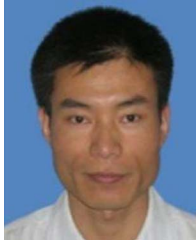
REFERENCES

- [1] G. Sohn and I. Dowman, "Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction," *ISPRS J. Photogramm. Remote Sens.*, vol. 62, no. 1, pp. 43–63, 2007.
- [2] J. Yuan, "Learning building extraction in aerial scenes with convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 11, pp. 2793–2798, Nov. 2018.
- [3] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1395–1403.
- [4] Y. Liu, M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3000–3009.
- [5] Y. Wang, X. Zhao, Y. Li, and K. Huang, "Deep crisp boundaries: From boundaries to higher-level tasks," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1285–1298, Mar. 2018.
- [6] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang, "Bi-directional cascade network for perceptual edge detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3828–3837.
- [7] X. S. Poma, E. Riba, and A. Sappa, "Dense extreme inception network: Towards a robust CNN model for edge detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 1923–1932.
- [8] Y. Cao, C. Lin, and Y. Li, "Learning crisp boundaries using deep refinement network and adaptive weighting loss," *IEEE Trans. Multimedia*, vol. 23, pp. 761–771, Apr. 2020.
- [9] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using brightness and texture," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 1–8.
- [10] X. Wen et al., "ME-Net: A multi-scale erosion network for crisp building edge detection from very high resolution remote sensing imagery," *Remote Sens.*, vol. 13, no. 19, Sep. 2021, Art. no. 3826.
- [11] T. Lu, D. Ming, X. Lin, Z. Hong, X. Bai, and J. Fang, "Detecting building edges from high spatial resolution remote sensing imagery using richer convolution features network," *Remote Sens.*, vol. 10, no. 9, Sep. 2018, Art. no. 1496.
- [12] S. Choi, J. Shin, K. Chi, and C. So, "Automated lineament extraction and edge linking using mask processing and Hough transform," in *Proc. KSRIS Conf.*, 1999, pp. 411–420.
- [13] A. Datta and S. K. Parui, "A robust parallel thinning algorithm for binary images," *Pattern Recognit.*, vol. 27, no. 9, pp. 1181–1192, 1994.
- [14] O. Hori and S. Tanigawa, "Raster-to-vector conversion by line fitting based on contours and skeletons," in *Proc. 2nd Int. Conf. Document Anal. Recognit.*, 1993, pp. 353–358.
- [15] J. Chen, Q. Lei, Y. Miao, and Q. Peng, "Vectorization of line drawing image based on junction analysis," *Sci. China Inf. Sci.*, vol. 58, no. 7, pp. 1–14, 2015.
- [16] I. Puhachov, W. Neveu, E. Chien, and M. Bessmeltsev, "Keypoint-driven line drawing vectorization via polyvector flow," *ACM Trans. Graph.*, vol. 40, no. 6, 2021, Art. no. 266.
- [17] H. P. Morevec, "Towards automatic visual obstacle avoidance," in *Proc. 5th Int. Joint Conf. Artif. Intell.*, 1977, Art. no. 584.
- [18] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 147–151.
- [19] S. M. Smith and J. M. Brady, "SUSAN—A new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45–78, May 1997.
- [20] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.
- [21] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, Jan. 2010.
- [22] M. Awrangjeb and G. Lu, "Robust image corner detection based on the chord-to-point distance accumulation technique," *IEEE Trans. Multimedia*, vol. 10, no. 6, pp. 1059–1072, Oct. 2008.
- [23] F. Mokhtarian and R. Suomela, "Robust image corner detection through curvature scale space," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1376–1381, Dec. 1998.
- [24] G. Xia, J. Huang, N. Xue, Q. Lu, and X. Zhu, "GeoSay: A geometric saliency for extracting buildings in remote sensing images," *Comput. Vis. Image Understanding*, vol. 186, pp. 37–47, 2019.
- [25] N. Xue, G. Xia, X. Bai, L. Zhang, and W. Shen, "Anisotropic-scale junction detection and matching for indoor images," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 78–91, Jan. 2018.
- [26] Z. Li, J. D. Wegner, and A. Lucchi, "Topological map extraction from overhead images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1715–1724.
- [27] Q. Li et al., "Instance segmentation of buildings using keypoints," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 1452–1455.
- [28] B. Chatterjee and C. Poullis, "Semantic segmentation from remote sensor data and the exploitation of latent learning for classification of auxiliary tasks," *Comput. Vis. Image Understanding*, vol. 210, 2021, Art. no. 103251.
- [29] J. Chen, C. Wang, H. Zhang, F. Wu, B. Zhang, and W. Lei, "Automatic detection of low-rise gable-roof building from single submeter SAR images based on local multilevel segmentation," *Remote Sens.*, vol. 9, no. 3, Mar. 2017, Art. no. 263.
- [30] Z. Hong, D. Ming, K. Zhou, Y. Guo, and T. Lu, "Road extraction from a high spatial resolution remote sensing image based on richer convolutional features," *IEEE Access*, vol. 6, pp. 46988–47000, 2018.
- [31] Z. Zhang and Y. Wang, "JointNet: A common neural network for road and building extraction," *Remote Sens.*, vol. 11, no. 6, Mar. 2019, Art. no. 696.
- [32] "Module: Measure.find_contours," Accessed: Jan. 27, 2022. [Online]. Available: https://scikit-image.org/docs/0.8.0/api/skimage.measure.find_contours.html
- [33] "Sklearn.decomposition.PCA," Accessed: Jan. 27, 2022. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>
- [34] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions," 2009. [Online]. Available: <https://authors.library.caltech.edu/27187/>
- [35] "sklearn.cluster.MeanShift," Accessed: Jan. 27, 2022. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html>
- [36] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation, Dept. Comput. Sci., Univ. Toronto, Toronto, Canada, 2013.
- [37] S. Saito and Y. Aoki, "Building and road detection from large aerial imagery," in *Proc. Image Process.: Mach. Vis. Appl. VIII*, 2015, pp. 153–164.
- [38] S. Saito, T. Yamashita, and Y. Aoki, "Multiple object extraction from aerial imagery with convolutional neural networks," *Electron. Imag.*, vol. 2016, no. 10, pp. 1–9, 2016.
- [39] J. Jing, C. Liu, W. Zhang, Y. Gao, and C. Sun, "ECFRNet: Effective corner feature representations network for image corner detection," *Expert Syst. Appl.*, vol. 211, 2023, Art. no. 118673.



Xiang Wen received the M.S. degree in photogrammetry and remote sensing from the Jiangsu Normal University, Xuzhou, China, in 2022. He is currently working toward the Ph.D. degree in geography with the Nanjing University, Nanjing, China.

His research interests include high-resolution remote sensing image processing, machine learning, and computer vision.



Xing Li received the M.S. degree in cartography and geographic information engineering from the Shandong University of Science and Technology, Qingdao, China, in 2006, and the Ph.D. degree in physical geography from the East China Normal University, Shanghai, China, in 2010.

He is currently an Associate Professor with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou, China. His research interests include high-resolution image processing and computer vision in environmental remote

sensing applications.



Wenquan Han received the Ph.D. degree in remote sensing of resources and environment from the Nanjing University, Nanjing, China, in 2012.

He is currently a Senior Engineer with the Nanjing Institute of Surveying, Mapping & Geotechnical Investigation, Co., Ltd., Nanjing, China. His research interests include image processing and 3-D model processing.



Erzhu Li received the M.S. degree in photogrammetry and remote sensing from the China University of Mining and Technology, Xuzhou, China, in 2014, and the Ph.D. degree in cartography and geographic information system from the Nanjing University, Nanjing, China, in 2017.

He is currently an Associate Professor with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou, China. His research interests include high-resolution image processing and computer vision in urban remote sensing

applications.



Wei Liu received the M.S. and Ph.D. degrees in cartography and geographic information engineering from the China University of Mining and Technology, Xuzhou, China, in 2007 and 2010, respectively.

He is currently an Associate Professor with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou, China. His research interests include spatial data quality checking, high-resolution remote sensing image processing, and GIS development and applications.



Lianpeng Zhang received the M.S. degree from the Shandong University of Science and Technology, Taian, China, in 1989, and the Ph.D. degree from the Shandong University of Science and Technology, Qingdao, China, in 2003, both in geodesy and surveying engineering.

He is currently a Professor with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou, China. His research interests include high-resolution image processing and computer vision in urban remote sensing applications.



Yihu Zhu received the B.E. degree in geomatics engineering from the Wuhan University, Wuhan, China, in 2005.

He is currently a Senior Engineer with the Jiangsu Geologic Surveying and Mapping Institute, Nanjing, China. His research interests include urban remote sensing applications and GIS development and applications.



Shengli Wang received the M.S. degree in photogrammetry and remote sensing from the Jiangsu Normal University, Xuzhou, China, in 2018.

He is currently an Engineer with the Jiangsu Geologic Surveying and Mapping Institute, Nanjing, China. His research interests include high-resolution image processing and computer vision.



Sibao Hao received the M.S. degree in cartography and geographic information system from the Nanjing Normal University, Nanjing, China, in 2007.

He is currently a Senior Engineer with the Jiangsu Geologic Surveying and Mapping Institute, Nanjing, China. His research interests include high-resolution image processing and GIS development and applications.