Tao Y, Zhang YT, Chen XJ. Element-arrangement context network for facade parsing. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 37(3): 652–665 May 2022. DOI 10.1007/s11390-022-2189-3

Element-Arrangement Context Network for Facade Parsing

Yan Tao (陶　琰), Yi-Teng Zhang (张翼腾), and
Xue-Jin Chen* (陈雪锦), *Senior Member, CCF, Member, ACM, IEEE*

National Engineering Laboratory for Brain-Inspired Intelligence Technology and Application, University of Science and Technology of China, Hefei 230026, China

E-mail: {ty990813, zytcc}@mail.ustc.edu.cn; xjchen99@ustc.edu.cn

Received Jan. 28, 2022; accepted April 24, 2022.

Abstract Facade parsing aims to decompose a building facade image into semantic regions of the facade objects. Considering each architectural element on a facade as a parameterized rectangle, we formulate the facade parsing task as object detection, allowing overlapping and nesting, which will support structural 3D modeling and editing for further applications. In contrast to general object detection, the spatial arrangement regularity and appearance similarity between the facade elements of the same category provide valuable context for accurate element localization. In this paper, we propose to exploit the spatial arrangement regularity and appearance similarity of facade elements in a detection framework. Our element-arrangement context network (EACNet) consists of two unidirectional attention branches, one to capture the column-context and the other to capture row-context to aggregate element-specific features from multiple instances on the facade. We conduct extensive experiments on four public datasets (ECP, CMP, Graz50, and eTRIMS). The proposed EACNet achieves the highest mIoU (82.1% on ECP, 77.35% on Graz50, and 82.3% on eTRIMS) compared with the state-of-the-art methods. Both the quantitative and qualitative evaluation results demonstrate the effectiveness of our dual unidirectional attention branches to parse facade elements.

Keywords facade parsing, element detection, layout regularity, spatial context

1 Introduction

Facade parsing aims to find regions of building facade components and annotate them with distinctive semantic categories (e.g., window, sill, balcony, and molding) in a given street-view facade image. This task potentially supports many real-world applications, especially for urban street reconstruction. However, facade parsing faces many challenges in natural urban scenes. Firstly, the facade style varies a lot among buildings. The diversity of texture and element structure makes it difficult to generate robust and accurate parsing results. Secondly, parsing a facade image may be more challenging due to shadows, illumination, perspective effect, and occlusions caused by cluttered objects. Most importantly, since the arrangement regularity of various building facade elements is naturally existing and widely presented, the parsing results should globally follow regular arrangement.

Facade parsing has been attracting lots of interest over the past few years. Traditional approaches usually combine architectural priors with image segmentation. The facade structural priors, such as element sizes, the spacing between elements, and hard alignment constraints, are encoded in the parsing procedure to introduce essential architectural information. Some grammar-based methods [1–5] perform top-down parsing procedures to model facades with predefined primitive shapes and grammar rules. Some other approaches [6–8] utilize the low-level information extracted by per-pixel classification to produce facade segmentation. Though these methods consider the facade regularity, they rely highly on hand-crafted knowledge priors. These hand-crafted structural constraints do not always fit individual facades, especially for complex scenes.

Regular Paper

Special Section of CVM 2022

This work was supported by the National Natural Science Foundation of China under Grant No. 61632006, and Tencent Corporation.

*Corresponding Author

Recent progress in deep learning and deep convolutional neural networks has made it possible to extract and utilize high-level features and global structural information of a building facade. Several learning-based methods [9–11] treat facade parsing as a semantic segmentation problem and employ popular convolutional neural networks (CNNs) to achieve better performance. DeepFacade [10, 11] illustrates the importance of facade structural priors and introduces the shape symmetry of facade elements as a constraint, by using bounding boxes as auxiliary data to refine the shape of segmentation regions. However, the element symmetry and the facade layout regularity, which are crucial for obtaining complete and reasonable facade parsing results, are ignored.

Though a pixel collection can flexibly describe freeform object shapes in the semantic segmentation framework, we argue that dense semantic region masks are not the most appropriate representation for facade parsing. First, objects on a facade usually appear as symmetrical quadrilaterals in a rectified street-view facade image. However, it is difficult to explicitly impose these geometric constraints directly in pixel-wise segmentation networks, while existing approaches add these constraints in loss functions [10, 11]. Second, facade segmentation usually results in a labeled mask image where each pixel is assigned a single category. However, facade components are not always disjoint. Overlapping frequently happens among various categories such as windows and blinds. Fig.1 shows a typical case where the balconies overlap with the bottom regions of their nearby windows. The dense single-category assignment makes the rendering and modeling of the overlapping regions much more complicated, even resulting in the structure loss of the nesting regions. In contrast, we propose a detection-based framework to decompose facade images while supporting overlapping facade elements and involving the global layout context to generate more regular facade arrangements.

The element layout usually presents a strong regularity and shows a grid-like element arrangement, as Fig.2 shows. A facade element is usually correlated with facade objects in the same row or column. For example, the window highlighted in Fig.2(a) can be accurately localized based on its related horizontal and vertical element groups though it is partially occluded by vegetation. Based on this observation, we leverage the spatial regularity of the facade layout in our element detection framework. We propose an element-arrangement context network (EACNet) to exploit the arrangement regularity among facade elements arranged in the same row or column. We conduct extensive experiments to evaluate the effectiveness of our method. Our EACNet achieves the top performance on the Graz50 [12] and ECP [3] datasets. Even on the challenging CMP [13] dataset, our EACNet effectively captures the element-arrangement spatial context and significantly facilitates the facade parsing task.

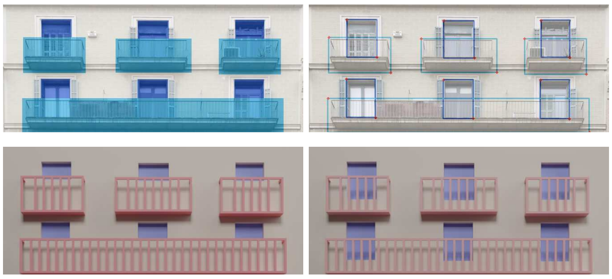

(a) (b)

Fig.1. Two representations of facade parsing (the upper layer) and modeling (the lower layer). (a) Semantic masks. (b) Bounding boxes. Our method aims to produce compact parameterized bounding boxes instead of dense pixel-wise semantics so that 3D facade models can be generated more efficiently while allowing structural overlapping of multiple elements.

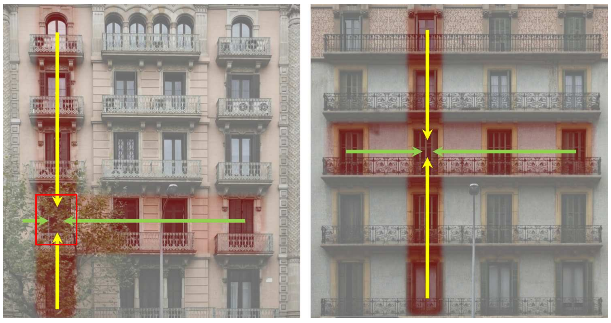

(a) (b)

Fig.2. Two examples demonstrating the spatial regularity of facades. The layout of a building facade presents strong regularity, as the architectural elements are well-aligned both vertically and horizontally. The spatial correlation between facade elements in the two directions provides the valuable context for the facade element detection.

2 Related Work

We discuss related work on traditional facade parsing and CNN-based facade segmentation. We also discuss several typical object detection approaches. In addition, we discuss attention mechanisms and several related general self-attention schemes.

Facade parsing and modeling from images have been extensively studied in computer graphics and computer vision. There are two mainstreams of traditional methods: utilizing grammar-based recognition and following conventional image segmentation pipeline. Grammar-based approaches model facades according to a set of parametric grammars, based on which the procedural modeling procedure can utilize image analysis techniques to derive a hierarchical facade subdivision from an image [14–16]. Similar ideas can be found in other methods that target general procedural modeling for structural objects [17–19]. The other stream of facade layout generation methods is segmenting the input images. Several approaches incorporate traditional machine learning to fit the procedural modeling pipeline [4, 20, 21]. Some others utilize architectural principles to optimize facade segmentation [5, 6, 8, 22, 23], aiming to produce more regular segmentation regions. In addition, the structural prior of facades including shape symmetry and layout regularity has also been demonstrated very effective to facilitate facade modeling from point clouds [2, 24] or structural facade editing [25].

With the rapid development of deep learning, CNN-based semantic segmentation frameworks have been adopted for facade parsing. Directly applying the fully-convolutional networks for semantic segmentation into facade segmentation [9, 26] generates pixel-wise label prediction. Subsequently, object symmetry is taken into account to refine the segmented region boundaries in DeepFacade [10] that uses a loss function to penalize segmentation regions that are not horizontally or not vertically symmetric. Its extension work [11] adds another loss term that forces the window regions to match the rectangular shapes obtained by a pre-trained auxiliary Mask R-CNN [27]. While DeepFacade methods [10, 11] focus on improving the regularity of the single element shape, our approach naturally ensures the single shape regularity and exploits global layout regularity with a well-designed attention scheme.

Object detection pipelines directly output rectangular boxes for objects in an image. Many two-stage region detection networks have been proposed [28, 29]. More recently, keypoint estimation has been utilized to locate objects for one-stage detection. CornerNet [30] detects objects by localizing a pair of key points and groups them by using associative embedding [31]. CenterNet [32] treats the object center as a single shape-agnostic anchor, detecting an object by extracting a center point, and thus needs not any keypoint grouping steps. Based on the one-stage detection framework, our EACNet is designed specifically for facade parsing by incorporating the spatial facade layout regularity.

Self-attention was first introduced in the pioneering work [33] to enhance the representation capability of neural networks and now is widely used for various tasks. However, self-attention suffers from quadratic computation and memory cost, which is particularly challenging for images. Recently, many efforts have been made to investigate sparse and memory-efficient forms, including hierarchical attention [34], clustering-based sparse attention [35], attention to sparse keypoints only [36], attention to image patches instead of pixels [37] and attention with linear complexity [38]. These methods can greatly reduce extra computation and memory costs and make self-attention more efficient.

For computer vision tasks, SENet [39] models channel-wise relationships in an attention mechanism. PSANet [40] learns two global attention maps to aggregate the contextual information for each position in the feature maps adaptively. The non-local network [41] generates a huge attention map by calculating pairwise affinities of all points in the feature maps. Transformer [33] is now the cutting-edge technology for modeling global relations and has been adopted for semantic segmentation. Segmenter [42] extends the patch-based transformer architecture [37] to the semantic segmentation problem for leveraging contextual information. Pyramid Vision Transformer [43] learns multi-scale patch embeddings through a progressive shrinking pyramid transformer architecture. However, the computation and memory cost for obtaining the attention maps for global contexts in these methods is significantly high. CCNet [44] develops a criss-cross attention module that captures contextual information in criss-cross paths instead of the whole image and then employs a recurrent operation to harvest full-image dependencies. Inspired by CCNet [44], we further decompose the criss-cross correlation into two independent unidirectional attention branches that only capture long-range dependencies between elements aligned in the same row and column separately, considering the spatial regularity of facade elements. This separation explicitly brings structural priors for the spatial correlation between pixels and makes our network more efficient and precise by considering the column-wise and row-wise distinction.

3 Our Approach

In this section, we first introduce the architecture of our EACNet. Then we describe the proposed element-arrangement context module (EACM) in detail, including the row and column context branches that capture the spatial context to enforce the arrangement regularity of the facade elements.

3.1 Network Architecture

Fig.3 shows the overview of the proposed EACNet. Given an input facade image, the Hourglass network[45] is employed as the backbone that downsamples the input image by four times and extracts feature maps $\boldsymbol{F}$ with the spatial dimension $H \times W$ from the input image. The feature maps $\boldsymbol{F} \in \mathbb{R}^{C\times H\times W}$ are then fed into EACM that learns the correlations between a position on the facade and all different positions in the same row and the same column. The long-range dependencies in the two axis-aligned directions are crucial for localizing facade objects because they show strong repetitiveness and alignment regularity in structure. The two branches in EACM produce feature maps $\boldsymbol{S}_{\mathrm{col}} \in \mathbb{R}^{C\times H\times W}$ and $\boldsymbol{S}_{\mathrm{row}} \in \mathbb{R}^{C\times H\times W}$ that collect spatial context in a single column and row, respectively. The feature maps $\boldsymbol{S}_{\mathrm{col}}$ and $\boldsymbol{S}_{\mathrm{row}}$ are concatenated and fed to a convolutional layer that acts as feature adaptation. The produced feature maps $\boldsymbol{M}$ are added to the image feature maps $\boldsymbol{F}$ to enhance the representation of each position. The enhanced feature maps $\boldsymbol{F'}$ are fed into a detector head to predict the bounding boxes that represent the parsing results.

3.2 Element-Arrangement Context Module

It is crucial to exploit the priors of facade structure such as shape symmetry and global alignment in facade parsing. To incorporate hand-crafted rules into an end-to-end neural network, existing CNN-based facade parsing methods either restrict the object shape by symmetry constraint[10, 11] or capture nonlocal contextual information. But they seldom take advantage of the holistic facade structure efficiently. Facade elements share strong repetitiveness and alignment in structure. To explicitly leverage the arrangement regularity, our EACM learns to exploit the correlations between the facade elements aligned in the same row and the same column. As shown in Fig.4, the proposed EACM contains a column-context branch and a row-context branch to collect element-arrangement spatial context in two directions. To the best of our knowledge, this is the first attempt to employ self-attention to incorporate the facade layout structural regularity into a facade parsing network.

Following the self-attention mechanism, we first apply three parallel convolutional layers with 1×1 filters on the feature maps $\boldsymbol{F}$ to obtain query feature maps $\boldsymbol{Q}$, key feature maps $\boldsymbol{K}$, and value feature maps $\boldsymbol{V}$, with dimensions of $C \times H \times W$. The two branches of EACM both use $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ to generate contextual features in two directions. For a pixel located at $\boldsymbol{p} = (i, j)$, the column-context branch calculates the correlations between $\boldsymbol{p}$ and all positions in the j-th column, and the row-context branch calculates the correlations between $\boldsymbol{p}$ and all positions in the i-th row. For a query vector $\boldsymbol{Q_p} \in \mathbb{R}^{C\times 1}$ in feature maps $\boldsymbol{Q}$, we extract key vectors from the feature maps $\boldsymbol{K}$ along the i-th row and the j-th column separately and compose two matrices from the two sets of feature vectors respectively:

$$\boldsymbol{X_p} = (\boldsymbol{K}_{(i,1)}, \boldsymbol{K}_{(i,2)}, \ldots, \boldsymbol{K}_{(i,j)}, \ldots, \boldsymbol{K}_{(i,W)}),$$
$$\boldsymbol{Y_p} = (\boldsymbol{K}_{(1,j)}, \boldsymbol{K}_{(2,j)}, \ldots, \boldsymbol{K}_{(i,j)}, \ldots, \boldsymbol{K}_{(H,j)}),$$

where $\boldsymbol{X_p} \in \mathbb{R}^{C\times 1\times W}$ and $\boldsymbol{Y_p} \in \mathbb{R}^{C\times H\times 1}$. In the column-context branch, the correlations between $\boldsymbol{p}$ and

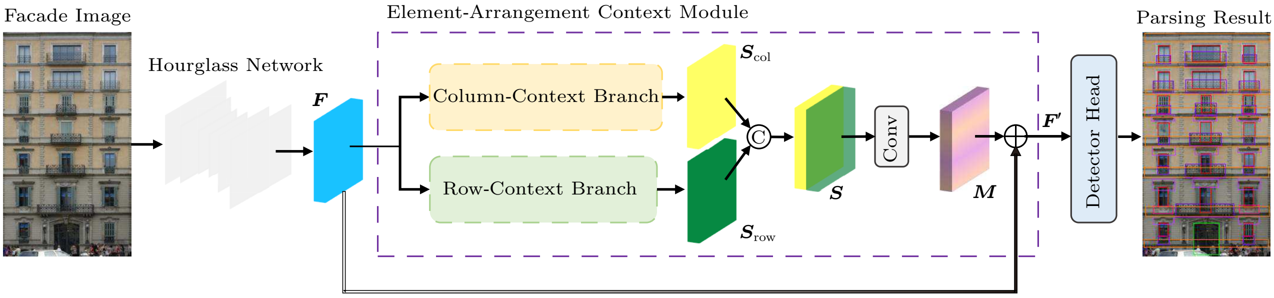


Fig. 3. Overview of the proposed EACNet. After extracting feature maps from the input image using an hourglass network, we aggregate the spatial context between facade elements by the proposed element-arrangement context module. Two rectilinear context branches are designed to capture the vertical and horizontal correlations between elements, and the contexts in the two directions are aggregated to enhance the local features. Finally, a detector head is attached to obtain the final facade parsing results from the aggregated feature maps. "©" denotes feature concatenation, and "⊕" denotes element-wise addition.

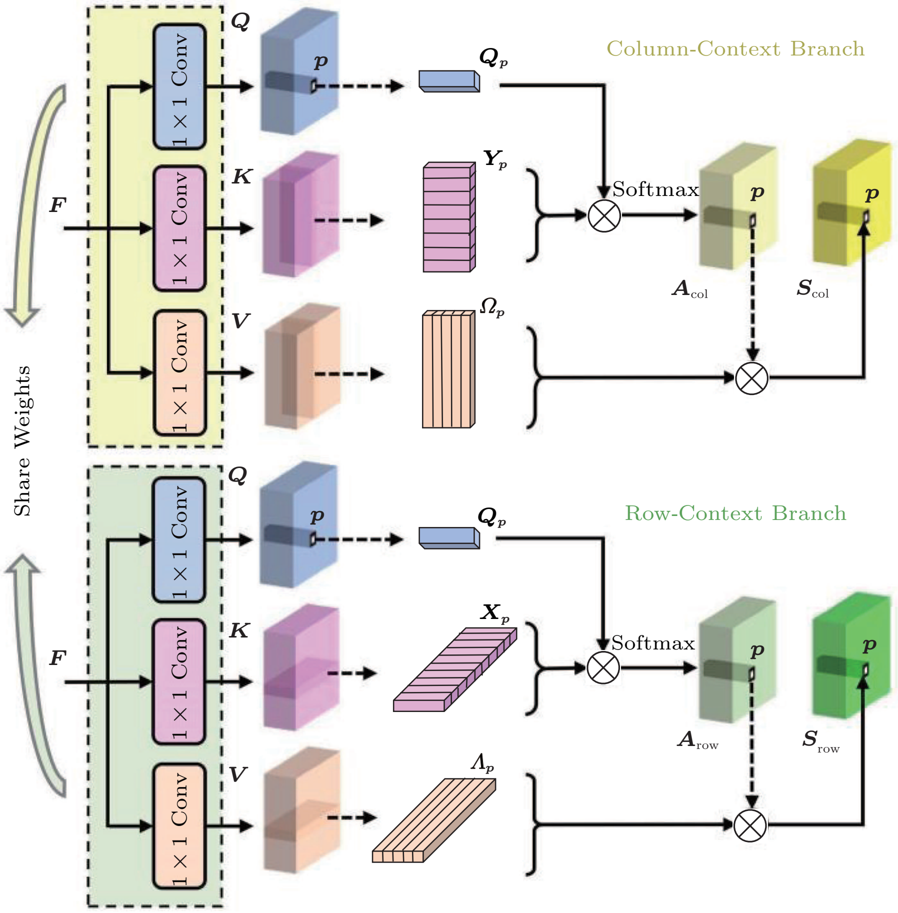


Fig.4. Architecture of the two context branches in EACM. The column-context branch and the row-context branch take feature maps produced by the backbone network to capture the spatial context in vertical and horizontal directions, respectively. "⊗" denotes matrix multiplication.

its corresponding column-path positions can be calculated and collected in a vector ${}^{\mathrm{c}}\boldsymbol{A}_{\boldsymbol{p}} \in \mathbb{R}^{H\times 1}$ located in attention maps $\boldsymbol{A}_{\mathrm{col}}$, which is defined as

$$ {}^{\mathrm{c}}\boldsymbol{A}_{\boldsymbol{p}}^{(k)} = \frac{\exp\left(\boldsymbol{Q}_{\boldsymbol{p}}^{\mathrm{T}}\boldsymbol{Y}_{\boldsymbol{p}}^{(k)}\right)}{\sum_{t=1}^{|\boldsymbol{Y}_{\boldsymbol{p}}|}\exp\left(\boldsymbol{Q}_{\boldsymbol{p}}^{\mathrm{T}}\boldsymbol{Y}_{\boldsymbol{p}}^{(t)}\right)}, $$

where ${}^{\mathrm{c}}\boldsymbol{A}_{\boldsymbol{p}}^{(k)}$ is the k-th element of vector ${}^{\mathrm{c}}\boldsymbol{A}_{\boldsymbol{p}}$, and $\boldsymbol{Y}_{\boldsymbol{p}}^{(k)}$ is the k-th feature vector of $\boldsymbol{Y}_{\boldsymbol{p}}$.

In the row branch, similar to the calculation of $\boldsymbol{A}_{\mathrm{col}}$, we calculate the attention maps $\boldsymbol{A}_{\mathrm{row}}$, where the vector located at position $\boldsymbol{p}$ is defined as

$$ {}^{\mathrm{r}}\boldsymbol{A}_{\boldsymbol{p}}^{(k)} = \frac{\exp\left(\boldsymbol{Q}_{\boldsymbol{p}}^{\mathrm{T}}\boldsymbol{X}_{\boldsymbol{p}}^{(k)}\right)}{\sum_{t=1}^{|\boldsymbol{X}_{\boldsymbol{p}}|}\exp\left(\boldsymbol{Q}_{\boldsymbol{p}}^{\mathrm{T}}\boldsymbol{X}_{\boldsymbol{p}}^{(t)}\right)}, $$

where ${}^{\mathrm{r}}\boldsymbol{A}_{\boldsymbol{p}}^{(k)}$ is the k-th element of vector ${}^{\mathrm{r}}\boldsymbol{A}_{\boldsymbol{p}} \in \mathbb{R}^{W\times 1}$, and $\boldsymbol{X}_{\boldsymbol{p}}^{(k)}$ is the k-th feature vector of $\boldsymbol{X}_{\boldsymbol{p}}$.

After obtaining attention maps $\boldsymbol{A}_{\mathrm{row}}$ and $\boldsymbol{A}_{\mathrm{col}}$ that measure correlations in row and column paths, we extract values from feature maps $\boldsymbol{V}$ in the row and column paths for further context aggregation. For a position $\boldsymbol{p} = (i, j)$, two matrices $\boldsymbol{\Lambda}_{\boldsymbol{p}}$ and $\boldsymbol{\Omega}_{\boldsymbol{p}}$ can be obtained. The c-th elements of $\boldsymbol{\Lambda}_{\boldsymbol{p}}$ and $\boldsymbol{\Omega}_{\boldsymbol{p}}$ are respectively defined as follows:

$$ \boldsymbol{\Lambda}_{\boldsymbol{p}}^{(c)} = (V_{ci1}, V_{ci2}, \ldots, V_{ciW})^{\mathrm{T}}, $$
$$ \boldsymbol{\Omega}_{\boldsymbol{p}}^{(c)} = (V_{c1j}, V_{c2j} \ldots, V_{cHj})^{\mathrm{T}}, $$

where V_{cij} denotes the value located at (i, j) of the c-th channel of the feature maps $\boldsymbol{V}$.

The elements of correlation vectors ${}^{\mathrm{c}}\boldsymbol{A}_{\boldsymbol{p}}$ and ${}^{\mathrm{r}}\boldsymbol{A}_{\boldsymbol{p}}$ are separately used as the weights of vectors $\boldsymbol{\Omega}_{\boldsymbol{p}}^{(c)}$ and $\boldsymbol{\Lambda}_{\boldsymbol{p}}^{(c)}$ for conducting spatial context aggregation at position $\boldsymbol{p}$, which generates ${}^{\mathrm{c}}\boldsymbol{S}_{\boldsymbol{p}} \in \mathbb{R}^{C\times 1}$ and ${}^{\mathrm{r}}\boldsymbol{S}_{\boldsymbol{p}} \in \mathbb{R}^{C\times 1}$ as follows:

$$ {}^{\mathrm{c}}\boldsymbol{S}_{\boldsymbol{p}}^{(c)} = {}^{\mathrm{c}}\boldsymbol{A}_{\boldsymbol{p}}^{\mathrm{T}}\boldsymbol{\Omega}_{\boldsymbol{p}}^{(c)}, $$
$$ {}^{\mathrm{r}}\boldsymbol{S}_{\boldsymbol{p}}^{(c)} = {}^{\mathrm{r}}\boldsymbol{A}_{\boldsymbol{p}}^{\mathrm{T}}\boldsymbol{\Lambda}_{\boldsymbol{p}}^{(c)}. $$

Collecting the spatial context at different positions finally gives the context feature maps $\boldsymbol{S}_{\mathrm{row}}$ and $\boldsymbol{S}_{\mathrm{col}}$, both with dimensions of $C \times H \times W$. They are concatenated and fused together by a convolution layer to produce the integrated contextual feature maps $\boldsymbol{M} \in \mathbb{R}^{C\times H\times W}$, which is then added to the image feature maps $\boldsymbol{F}$ to produce the enhanced feature maps $\boldsymbol{F'}$ as

$$\boldsymbol{F'} = \omega(\boldsymbol{S}) + \boldsymbol{F},$$

where ω is a projection function implemented by a convolutional layer with 1×1 kernel size. $\boldsymbol{S} \in \mathbb{R}^{2C\times H\times W}$ is produced by concatenating $\boldsymbol{S}_{\text{col}}$ and $\boldsymbol{S}_{\text{row}}$ together.

3.3 Detector Head

The enhanced feature maps $\boldsymbol{F'}$ are fed into a detector head to obtain the final facade parsing results. As shown in Fig.5, it is effective to represent a symmetric facade element by a center and its width and height. We employ CenterNet[32] which models an object bounding box with a center point and an object size in our EACNet. The detector head consists of three branches. Each branch applies convolutional layers on $\boldsymbol{F'}$ to generate a set of heatmaps for center location prediction for each element category, local offset prediction, and object size prediction, respectively. The center location prediction branch generates $\hat{\boldsymbol{E}} \in \mathbb{R}^{C'\times H\times W}$, where C' is the number of categories of the facade elements. The value of $\hat{E}_{cij}$ at location $\boldsymbol{p} = (i, j)$ is the score for class c in the predicted heatmaps. The local offset prediction branch generates $\hat{\boldsymbol{O}} \in \mathbb{R}^{2\times H\times W}$, which is used as a slight adjustment of the center location. The object size prediction branch generates $\hat{\boldsymbol{U}} \in \mathbb{R}^{2\times H\times W}$, which gives the height and the width of an object. To obtain the coordinates of center points, a 3×3 max pooling layer is applied on $\hat{\boldsymbol{E}}$ for peaks extraction.

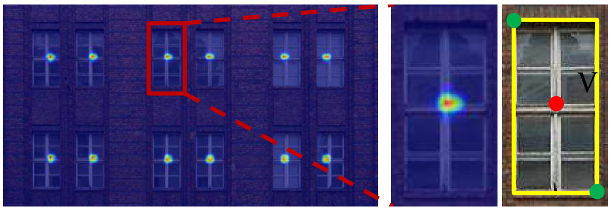


Fig.5. Output of the detector head in EACNet. We encode a facade object by a center point and its size parameters and predict heatmaps for the center point.

During training, we use a pixel-wise logistic regression with focal loss[46] for center location prediction:

$$L_{\boldsymbol{p}} = -\frac{1}{N}\sum_{c,i,j}\begin{cases}(1-\hat{E}_{cij})^{\alpha}\log(\hat{E}_{cij}), \\ \qquad \text{if } E_{cij} = 1, \\ (1-E_{cij})^{\beta}(\hat{E}_{cij})^{\alpha}\log(1-\hat{E}_{cij}), \\ \qquad \text{otherwise,}\end{cases}$$

where E is the ground truth and N is the number of facade objects. Both α and β are hyper-parameters that control the contribution of each point. We set $\alpha = 2$ and $\beta = 4$ in all experiments. The local offset and the object size are both trained with the loss function $L_1(\cdot,\cdot)$ that computes the L1 distance between the ground truth and the predicted values. The total loss function is

$$L = L_{\boldsymbol{p}} + \frac{\lambda}{N}\sum_{k}^{N} L_1(\hat{O}_k, O_k) + \frac{\mu}{N}\sum_{k}^{N} L_1(\hat{U}_k, U_k), \quad (1)$$

where O_k and U_k are the location offset and the object size of the k-th element, respectively. The scale factors λ and μ are used for weight adjustment.

4 Experimental Results and Discussions

In this section, we first introduce four facade datasets used for evaluation and present the corresponding evaluation metrics and the training details. Then, we compare our facade parsing method with existing segmentation-based facade parsing methods[6, 8, 10, 11, 20, 22, 23, 47, 48]. A series of ablation experiments are also conducted to demonstrate the effectiveness of the proposed EACM.

4.1 Datasets and Evaluation Metrics

Four public facade datasets are used in our experiments, including ECP[3], CMP[13], Graz50[12], and eTRIMS[11]. The first three contain rectified facade images with their semantic label masks. Images in the eTRIMS dataset are not rectified.

The ECP facade dataset[3] consists of 104 well-rectified building facade images. All the images contain facades from Paris and share similar architectural styles. The pixel annotations contain eight classes, including "window", "wall", "balcony", "door", "shop", "sky", "chimney", and "roof". Since there are some categories not belonging to facade elements, we choose "window", "balcony", "door", and "shop" for evaluation. Since the ECP dataset does not have bounding box annotation, we perform contour fitting on the provided semantic masks to generate the bounding box for each element. For overlapping elements, we adjust their bounding box sizes to match the corresponding regions. We follow DeepFacade-V2[11] to divide the dataset, using 80 images for training and 24 for testing.

The Graz50 facade dataset[12] contains 50 facade images with multiple building styles. Similar to the

ECP dataset, the Graz50 dataset only contains rectangular areas labeled as ground truth semantic masks. Contour fitting is also applied to obtain suitable bounding box annotation. The provided data contains two facade element classes, "window", and "door", which are both used in our experiments. We follow the dataset division strategy used in DeepFacade-V2 [11], using 30 images for training and 20 images for testing.

The CMP facade dataset [13] contains 606 rectified images of facades with diverse architectural styles. The dataset is split into two parts that consist of 378 and 228 images. The latter part contains more irregular and non-planar facades that often have substantial occlusion from vegetation, making the CMP dataset very challenging. The annotation of this dataset is a set of rectangles with class labels and allows overlapping and nesting. The dataset includes 12 specified classes. In our experiment, we use six categories that belong to facade elements, including "sill", "balcony", "door", "molding", "window", and "cornice". We use 484 images that are randomly selected from two subsets for training. The remaining 122 facade images are used for testing.

4.2 Training Settings

The Hourglass backbone [45] used in our EACNet is initialized using the weights of a model pre-trained on the COCO dataset [49]. The remaining part of the network is initialized randomly. In all experiments, the network is trained on a single GPU, using an Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the initial learning rate as 0.000 1, 0.000 2, 0.000 2, and 0.000 4 for the CMP, ECP, eTRIMS, and Graz50 datasets, respectively. The scale factors λ and μ in (1) are set to 1 and 0.1 respectively. We use the batch size 4 for all the four datasets. For the ECP, eTRIMS, and Graz50 datasets, the network is trained for 120, 100, and 80 epochs respectively. For the CMP dataset, the network is trained for 200 epochs, and we reduce the learning rate by 90% after 140 epochs. We use random horizontal flipping, random scaling in the range of $[0.6, 1.3]$, and color jittering for data augmentation. We randomly crop large images or pad small images into a fixed size for training. For the ECP, CMP, and eTRIMS datasets, we train on an input resolution of 512×512. Because the images in Graz50 have lower resolutions, which vary between 200 and 500 pixels in the height and the width respectively, we use 256×256 input resolution for this dataset.

4.3 Quantitative Evaluation

We quantitatively evaluate our method by comparing it with several facade parsing approaches [10, 11, 20, 22, 23, 47, 48] on the Graz50 and ECP datasets. As stated in the recent work DeepFacade-V2 [11], most of the existing approaches merely use the simple pixel accuracy metric for evaluation. However, the high pixel accuracy does not always imply superior performance because of the class imbalance. Following the previous work [11], we mainly use the intersection over union (IoU) metric for evaluation and also report pixel accuracy results as reference.

Table 1 shows the performance of our method and other state-of-the-art methods on the Graz50 dataset. The bold numbers indicate the highest value for each metric. As it shows, our method achieves the highest average pixel accuracy. Compared with the state-of-the-art method DeepFacade-V2 [11], our method gives better IoU results by a large margin.

Table 1. Quantitative Comparison on the Graz50 Dataset

Method	Pixel Accuracy (%)			IoU (%)	
	Window	Door	Avg.	Window	Door
Koziński *et al.* [47]	82.0	50.0	66.0	–	–
Koziński *et al.* [22]	84.0	60.0	72.0	–	–
Cohen *et al.* [23]	85.0	64.0	74.5	–	–
Rahmani *et al.* [20]	79.3	79.1	79.2	–	–
DeepFacade-V1 [10]	87.7	88.2	87.9	–	–
Rahmani *et al.* [48]	83.7	**93.8**	88.8	–	–
DeepFacade-V2 [11]	88.8	89.1	88.9	71.3	56.5
Ours	**89.9**	87.8	**88.9**	**80.9**	**73.8**

In Table 2 and Table 3, we provide the quantitative comparison on the ECP dataset. It shows that our method outperforms the state-of-the-art method in IoU of all classes and provides comparable pixel accuracy results with DeepFacade-V2 [11]. "Δ" denotes the performance gain brought by our EACNet compared with DeepFacade-V2, showing the superiority of our approach.

Table 1, Table 2, and Table 3 show that our method provides a much higher IoU on each facade element category, especially those highly aligned and repetitive in structure. In particular, for the "window" category which is the most frequent element on facades, compared with DeepFacade-V2 [11], our method improves the IoU by about 10% on both the Graz50 and ECP datasets. It demonstrates that our EACNet effectively leverages the layout regularity of building facades and exploits long-range dependencies between facade elements.

Table 2. Comparison of Pixel Accuracy on the ECP Dataset

Method	Window	Balcony	Door	Shop	Avg.
Cohen *et al.* [6]	85.0	91.0	79.0	94.0	87.3
ATLAS [8]	78.0	87.0	71.0	95.0	82.8
Cohen *et al.* [23]	87.0	92.0	79.0	96.0	88.5
Rahmani *et al.* [20]	80.4	86.4	79.5	95.2	85.4
DeepFacade-V1 [10]	93.0	95.0	90.9	95.6	93.6
Rahmani *et al.* [48]	78.6	89.2	89.2	**96.3**	88.3
DeepFacade-V2 [11]	**97.6**	**96.2**	92.3	96.0	**95.5**
Ours	94.4	95.9	**95.3**	92.0	94.4

Table 3. Comparison of IoU on the ECP Dataset

Method	Window	Balcony	Door	Shop	Avg.
DeepFacade-V2 [11]	80.3	85.2	63.1	80.3	77.2
Ours	**89.8**	**88.0**	**64.3**	**86.1**	**82.1**
Δ	9.5	2.8	1.2	5.8	4.9

4.4 Qualitative Evaluation

To better demonstrate the superiority of our facade parsing framework, we show some facade parsing results in Fig.6 of our method and the state-of-the-art methods DeepFacade-V1/V2 on the ECP dataset. DeepFacade-V1 tends to produce rough region boundaries for facade elements. DeepFacade-V2 produces more rectangular regions but mistakenly classifies the door as a window in the first row. In contrast, our method produces more regular regions for various facade element categories. Moreover, the parameterized parsing results allow overlapping and nesting, thus being more applicable than dense pixel-wise masks to applications such as facade modeling. In particular, though the area where windows and balconies overlap has a complex texture, our parsing framework is able to produce complete regions for “window” and “balcony” objects.

4.5 Results on Unrectified Facade Images

While the results on the ECP and Graz50 datasets well demonstrate the effectiveness of our EACNet, we also show the flexibility of our EACNet on parsing unrectified facade images. On unrectified facade images, elements are not perfectly rectangular, for which our EACNet is not applicable directly. However, there are many well-established rectification approaches for facade images. We take the TILT approach [50] which estimates the homography matrix for image rectification based on low-rank texture features. Given an image region that contains windows, TILT [50] estimates a homography matrix and applies the projection transformation on the entire image to produce a rectified facade image.

We conduct evaluations on the 8-class eTRIMS datasets [11], which contain 60 facade images from diffe-

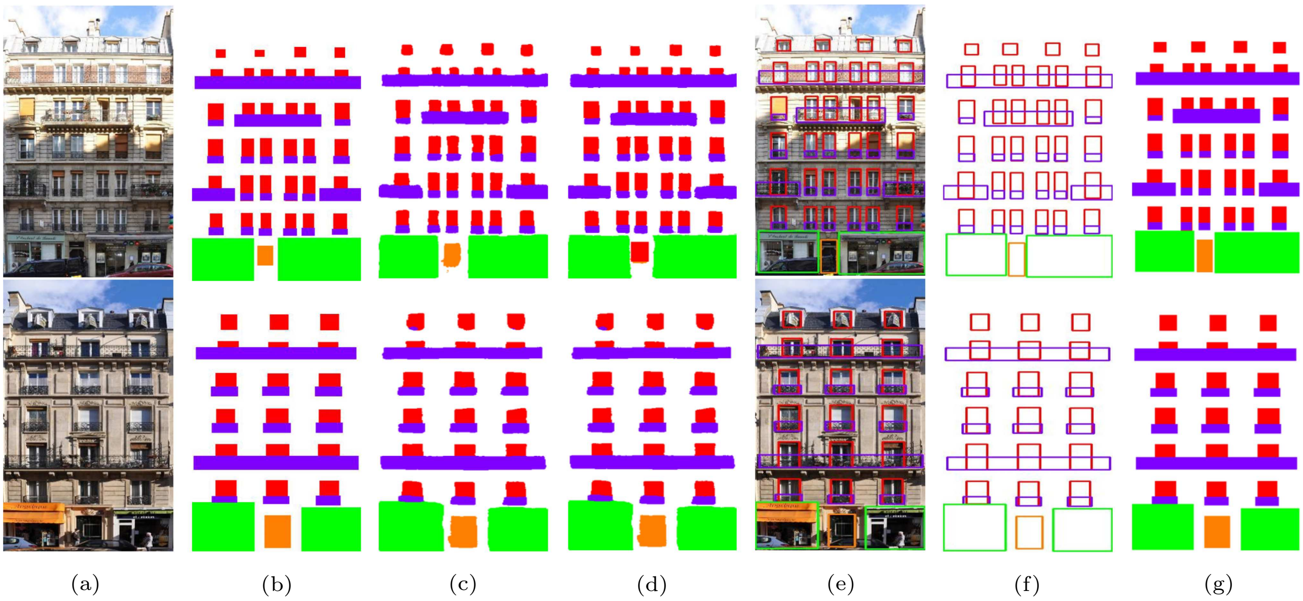


Fig.6. Qualitative comparisons of our method and the state-of-the-art facade parsing methods DeepFacade-V1 [10] and DeepFacade-V2 [11] on the ECP dataset. (a) Input image. (b) Ground-truth annotation. (c) Segmentation result of DeepFacade-V1 [10]. (d) Segmentation result of DeepFacade-V2 [11]. (e) Detection results of our EACNet mapped on the input image. (f) Detected bounding boxes. (g) Semantic masks rendered from the detection results of our EACNet.

rent perspectives. We use 48 images for training and 12 images for testing. The eTRIMS dataset consists of eight classes including "window", "wall", "door", "sky", "pavement", "vegetation", "car", and "road". Putting aside the categories not belonging to facade elements, we choose the "window" and "door" categories for evaluation. Since only semantic masks on the unrectified views are provided in the eTRIMS dataset, we manually label the bounding boxes for evaluation.

We train two models of our EACNet on the rectified images and the unrectified images respectively and compare the results with those of the DeepFacade-V2 [11] in Table 4. The "Ours-Unrectified" model is directly trained with the 2D bounding boxes of facade elements on unrectified perspectives. It is reasonable that this model cannot achieve a higher IoU with the ground-truth segmentation masks that are not rectangular. "Ours-Rectified" is the model trained with 2D bounding boxes that are well-fitted to the element regions on the rectified images. We test our model on the testing set of the rectified images and obtain the bounding boxes detected on the rectified images. Then we apply the inverse projection transformation on the bounding boxes to produce the semantic masks on the unrectified images. We calculate the IoU with the ground truth masks. As Table 4 shows, our method outperforms the segmentation-based method DeepFacade-V2 [11] by a large margin with image rectification.

Table 4. Comparison of IoU on the eTRIMS Dataset [11]

Method	Window	Door	Avg.
DeepFacade-V2 [11]	71.1	77.9	74.5
Ours-Unrectified	65.2	68.8	67.0
Ours-Rectified	**85.2**	**79.4**	**82.3**

Fig.7 shows an example of facade parsing for unrectified images using different methods. Though the segmentation-based method is flexible to represent non-rectangular regions under perspective projection, it fails to generate accurate and regular region boundaries for facade elements. Due to the restriction of rectangular shapes of the detection framework, directly applying our EACNet on the unrectified images successfully detects all elements but fails to generate well-fitted region boundaries. In comparison, with a well-established rectification step, our EACNet can produce accurate and structured region boundaries for facade elements.

4.6 Ablation Study

To verify the rationality of the proposed EACM, we carry out ablation experiments mainly on the challenging CMP dataset. ECP and Graz50 are also used for conducting additional comparisons.

4.6.1 Effect of EACM

In Table 5, we show the quantitative performance of our method with different configurations on the CMP dataset. "+ EACM" means adding an EACM between the Hourglass backbone and the detector head. "Flip Test" means combining horizontally flipped images during inference, which is widely used in recent detection networks [30, 32]. "✓" means using corresponding configurations mentioned above for experiments. We use the average precision over all the IoU thresholds (AP), the average precision at IoU threshold 0.5 (AP^{50}), and 0.75 (AP^{75}) for evaluation. The last six columns in Table 5 are per-class AP results of facade element categories of the CMP dataset. The results show that our EACM consistently improves the three AP metrics and per-class AP of important facade element classes. In particular, our EACM significantly improves the

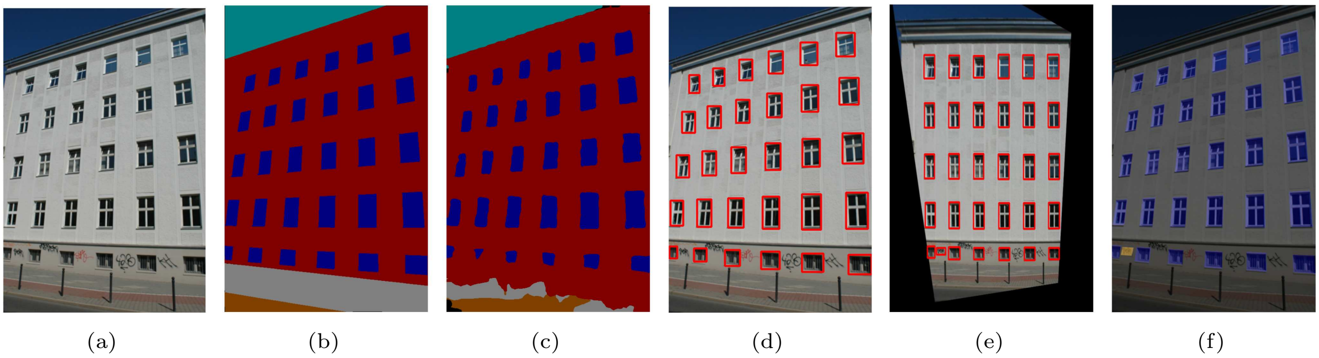


Fig.7. Qualitative comparison for unrectified facade images. (a) Input image. (b) Ground-truth masks. (c) Semantic segmentation result of DeepFacade-V1 [10]. (d) Our detection results without rectification. (e) Our detection results on the rectified image. (f) Semantic masks obtained by transforming the detection results on the rectified image to the original view.

Table 5. Effect of EACM

Flip Test	+EACM	AP	AP^{50}	AP^{75}	Sill	Balcony	Door	Molding	Window	Cornice
		38.6	67.1	38.9	38.6	33.9	33.8	25.3	59.2	40.8
	✓	39.2	67.5	40.1	39.9	35.0	34.4	24.9	60.4	40.6
✓		39.7	67.9	41.0	40.3	34.9	34.5	25.5	60.8	**42.2**
✓	✓	**40.2**	**68.4**	**42.3**	**40.9**	**35.8**	**34.6**	**26.0**	**62.0**	41.6

parsing accuracy of the "window" category (from 59.2 to 60.4 without flip-test and from 60.8 to 62.0 with flip-test). It is mainly because that windows show strong regularity and repetitiveness and our EACM effectively exploits the arrangement regularity and appearance similarity among window elements. Doors do not strictly follow the arrangement regularity with other elements. Nevertheless, the slight improvement for the "door" category also indicates that our EACM is also helpful for shape regularity since it collects local spatial context for each position on a door.

4.6.2 Different Attention Mechanisms

As described in Subsection 3.2, our EACM is designed to collect row-column spatial contextual information and leverage the arrangement regularity of the facade elements. The recurrent criss-cross attention (RCCA) module of CCNet[44] collects spatial context in criss-cross paths, which is similar to but different from our EACM. We compare our EACM with RCCA on the Graz50, ECP, and CMP datasets. We replace EACM with RCCA in our framework and use the same training settings for comparison. We test two models, $\mathbf{RCCA^1}$ and $\mathbf{RCCA^2}$ which employ one-loop and two-loops of RCCAs, respectively. As Table 6 shows, our EACM brings performance gain on all three datasets, while RCCA only achieves slight improvement on the CMP dataset. RCCA even performs worse than the baseline network on Graz50 and ECP. One reason is that RCCA collects the contextual information from all the pixels on the criss-cross paths and applies softmax on them, and subsequently cannot efficiently utilize the element-arrangement regularity on each direction separately. As a result, for the Graz50 and ECP datasets that contain facades with neatly arranged facade elements, RCCA does not work well. For facades with complex layouts and more categories in CMP, the dense full-image contextual information harvested by RCCA can be helpful. In contrast, our EACM effectively exploits the layout regularity in horizontal and vertical directions separately and outperforms RCCA on various scenarios.

Table 6. Comparison of the Proposed EACM and RCCA[44]

Dataset	Method	AP	AP^{50}	AP^{75}
Graz50	Baseline	65.8	94.1	**85.2**
	+RCCA^1	62.3	94.1	79.7
	+RCCA^2	63.8	94.7	83.1
	+EACM	**68.2**	**96.8**	84.2
ECP	Baseline	79.3	99.4	93.6
	+RCCA^1	78.1	99.4	93.0
	+RCCA^2	78.4	99.4	94.1
	+EACM	**80.1**	**99.4**	**95.2**
CMP	Baseline	39.7	67.9	41.0
	+RCCA^1	39.7	68.4	40.7
	+RCCA^2	39.8	68.3	41.2
	+EACM	**40.2**	**68.4**	**42.3**

4.6.3 Visualization of EACM

To validate the effectiveness of the proposed EACM on leveraging the layout regularity, we visualize the attention maps in Fig. 8. We can see that EACM focuses on element regions aligned in the same row or column, which proves that our method effectively exploits the spatial arrangement regularity and appearance similarity. In addition, we further investigate the effect of EACM by exploring two different strategies for fusing the contextual features produced by the two context branches. Besides concatenating feature maps $\boldsymbol{S}_{\mathrm{row}}$ and $\boldsymbol{S}_{\mathrm{col}}$ to produce $\boldsymbol{S}$, the other fusion strategy is element-wise addition. The precision-recall curves under different IoU thresholds are shown in Fig. 9. The results indicate that the two configurations of EACM both make performance improvement and concatenation fusion achieves the best performance.

4.6.4 Different Backbones

To further demonstrate the effectiveness of our EACM on various networks, we combine our EACM with different backbone networks, including ResNet-101[51], DLA-34[52], and Hourglass[45]. Table 7 shows our quantitative results on the CMP dataset. "+EACM" means adding our EACM module between the backbone network and the detector head. "Flip" means using test-time flip augmentation. "w/o Flip"

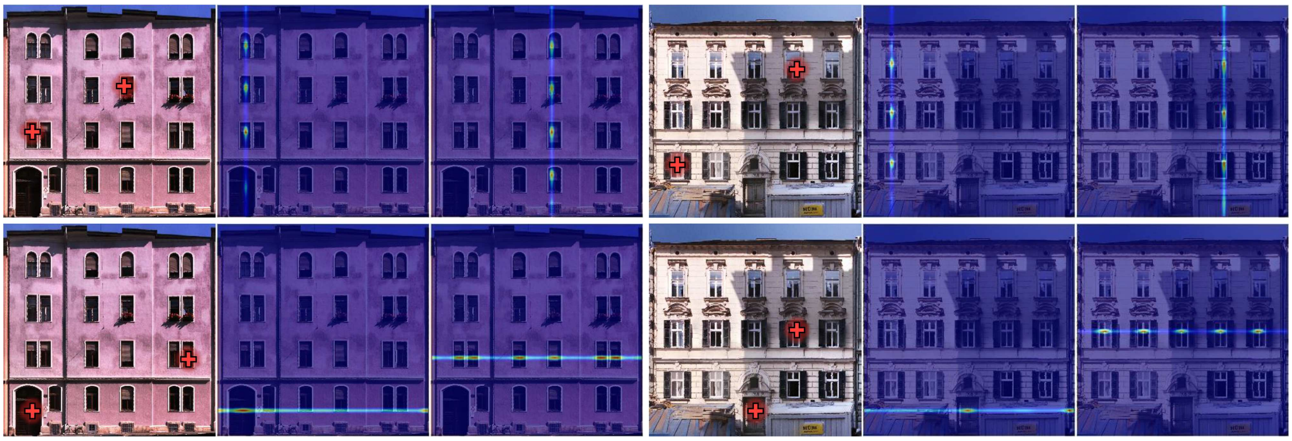
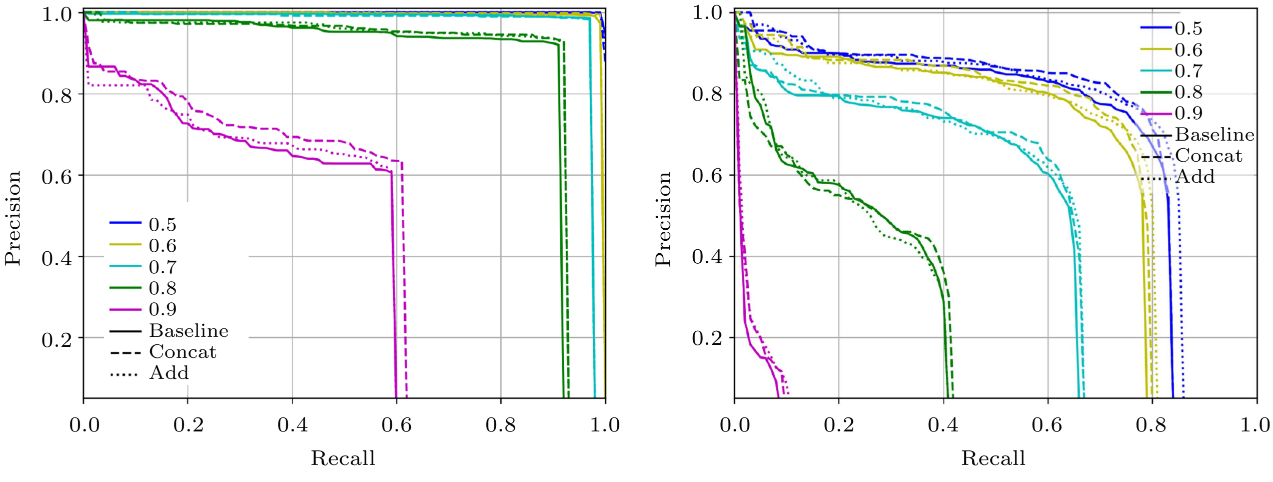


Fig.8. Visualization of attention maps $\boldsymbol{A}_{\text{col}}$ and $\boldsymbol{A}_{\text{row}}$ in our EACM. The query points are marked in red.



(a) (b)

Fig.9. Effect of using different fusion strategies in EACM. We show precision-recall curves under different IoU thresholds. (a) Results on the ECP dataset. (b) Results on the CMP dataset. The solid lines correspond to the baseline. The dashed lines and the dotted lines are results of concatenation (concat) and addition (add) fusion respectively.

Table 7. Comparison of Different Backbone Networks on the CMP Dataset

Backbone	+EACM	AP		AP^{50}		AP^{75}	
		w/o Flip	Flip	w/o Flip	Flip	w/o Flip	Flip
Hourglass		38.6	39.7	67.1	67.9	38.9	41.0
	✓	**39.2**	**40.2**	**67.5**	**68.4**	**40.1**	**42.3**
DLA-34		32.4	33.7	62.8	63.7	29.8	31.2
	✓	33.5	34.6	63.8	65.1	31.7	33.0
ResNet-101		29.9	31.0	60.8	61.9	26.7	27.7
	✓	30.9	31.9	62.4	63.8	26.8	28.2

means no flip test-time augmentation. From the reported AP, AP^{50}, and AP^{75} metrics, we can see that adding our EACM brings performance gains on all the three backbone networks, demonstrating that our EACM facilitates the facade parsing task by exploiting the spatial arrangement regularity and appearance similarity of facade elements.

5 Conclusions

In this paper, we introduced an element-arrangement context network, EACNet, for facade parsing by representing facade elements with parameterized axis-aligned bounding boxes. Our element-arrangement context module, EACM, collects spatial column-context and row-context, effectively leveraging the spatial arrangement regularity and appearance similarity. Experimental results on four public datasets showed that our facade parsing framework outperformed the state-of-the-art methods. The structured box outputs of facade elements would facilitate subsequent facade modeling applications. In the future, we would like to extend our facade parsing approach to multi-view street images and produce more completed and structured facade models for large-scale city scenes.

References

[1] Müller P, Zeng G, Wonka P, van Gool L. Image-based procedural modeling of facades. *ACM Transactions on Graphics*, 2007, 26(3): Article No. 85. DOI: 10.1145/1276377.1276484.

[2] Shen C H, Huang S S, Fu H B, Hu S M. Adaptive partitioning of urban facades. *ACM Transactions on Graphics*, 2011, 30(6): Article No. 184. DOI: 10.1145/2070781.2024218.

[3] Teboul O, Simon L, Koutsourakis P, Paragios N. Segmentation of building facades using procedural shape priors. In *Proc. the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp.3105-3112. DOI: 10.1109/CVPR.2010.5540068.

[4] Teboul O, Kokkinos I, Simon L, Koutsourakis P, Paragios N. Shape grammar parsing via reinforcement learning. In *Proc. the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, June 2011, pp.2273-2280. DOI: 10.1109/CVPR.2011.5995319.

[5] Yang C, Han T, Quan L, Tai C L. Parsing façade with rank-one approximation. In *Proc. the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp.1720-1727. DOI: 10.1109/CVPR.2012.6247867.

[6] Cohen A, Schwing A G, Pollefeys M. Efficient structured parsing of facades using dynamic programming. In *Proc. the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp.3206-3213. DOI: 10.1109/CVPR.2014.410.

[7] Martinović A, Mathias M, Weissenberg J, van Gool L. A three-layered approach to facade parsing. In *Proc. the 12th European Conference on Computer Vision*, Oct. 2012, pp.416-429. DOI: 10.1007/978-3-642-33786-4_31.

[8] Mathias M, Martinović A, van Gool L. ATLAS: A three-layered approach to facade parsing. *International Journal of Computer Vision*, 2016, 118(1): 22-48. DOI: 10.1007/s11263-015-0868-z.

[9] Schmitz M, Mayer H. A convolutional network for semantic facade segmentation and interpolation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2016, XLI-B3: 709-715. DOI: 10.5194/isprs-archives-XLI-B3-709-2016.

[10] Liu H, Zhang J, Hoi S C H. DeepFacade: A deep learning approach to facade parsing. In *Proc. the 26th International Joint Conference on Artificial Intelligence*, Aug. 2017, pp.2301-2307. DOI: 10.24963/ijcai.2017/320.

[11] Liu H, Xu Y, Zhang J, Zhu J, Li Y, Hoi S C H. DeepFacade: A deep learning approach to facade parsing with symmetric loss. *IEEE Transactions on Multimedia*, 2020, 22(12): 3153-3165. DOI: 10.1109/TMM.2020.2971431.

[12] Riemenschneider H, Krispel U, Thaller W, Donoser M, Havemann S, Fellner D, Bischof H. Irregular lattices for complex shape grammar facade parsing. In *Proc. the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp.1640-1647. DOI: 10.1109/CVPR.2012.6247857.

[13] Tyleček R, Šára R. Spatial pattern templates for recognition of objects with regular structure. In *Proc. the 35th German Conference on Pattern Recognition*, Sept. 2013, pp.364-374. DOI: 10.1007/978-3-642-40602-7_39.

[14] Bao F, Schwarz M, Wonka P. Procedural facade variations from a single layout. *ACM Transactions on Graphics*, 2013, 32(1): Article No. 8. DOI: 10.1145/2421636.2421644.

[15] Dang M, Ceylan D, Neubert B, Pauly M. SAFE: Structure-aware facade editing. *Computer Graphics Forum*, 2014, 33(2): 83-93. DOI: 10.1111/cgf.12313.

[16] Ilčík M, Musialski P, Auzinger T, Wimmer M. Layer-based procedural design of façades. *Computer Graphics Forum*, 2015, 34(2): 205-216. DOI: 10.1111/cgf.12553.

[17] Han F, Zhu S C. Bottom-up/top-down image parsing by attribute graph grammar. In *Proc. the 10th IEEE International Conference on Computer Vision*, Oct. 2005, pp.1778-1785. DOI: 10.1109/ICCV.2005.50.

[18] Talton J O, Lou Y, Lesser S, Duke J, Měch R, Koltun V. Metropolis procedural modeling. *ACM Transactions on Graphics*, 2011, 30(2): Article No. 11. DOI: 10.1145/1944846.1944851.

[19] Yeh Y T, Breeden K, Yang L, Fisher M, Hanrahan P. Synthesis of tiled patterns using factor graphs. *ACM Transactions on Graphics*, 2013, 32(1): Article No. 3. DOI: 10.1145/2421636.2421639.

[20] Rahmani K, Huang H, Mayer H. Facade segmentation with a structured random forest. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2017, IV-1/W1: 175-181. DOI: 10.5194/isprs-annals-IV-1-W1-175-2017.

[21] Gaddle R, Jampani V, Marlet R, V Gehler P. Efficient 2D and 3D facade segmentation using auto-context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, 40(5): 1273-1280. DOI: 10.1109/TPAMI.2017.2696526.

[22] Koziński M, Gadde R, Zagoruyko S, Obozinski G, Marlet R. A MRF shape prior for facade parsing with occlusions. In *Proc. the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp.2820-2828. DOI: 10.1109/CVPR.2015.7298899.

[23] Cohen A, R Oswald M, Liu Y, Pollefeys M. Symmetry-aware façade parsing with occlusions. In *Proc. the 2017 International Conference on 3D Vision*, Oct. 2017, pp.393-401. DOI: 10.1109/3DV.2017.00052.

[24] Nan L, Sharf A, Zhang H, Cohen-Or D, Chen B. SmartBoxes for interactive urban reconstruction. *ACM Trans. Graph.*, 2010, 29(4): Article No. 93. DOI: 10.1145/1778765.1778830.

[25] Zhang H, Xu K, Jiang W, Lin J, Cohen-Or D, Chen B. Layered analysis of irregular facades via symmetry maximization. *ACM Trans. Graph.*, 2013, 32(4): Article No. 121. DOI: 10.1145/2461912.2461923.

[26] Femiani J, Reyaz Para W, Mitra N, Wonka P. Facade segmentation in the wild. arXiv:1805.08634, 2018. https://arxiv.org/pdf/1805.08634.pdf, Jan. 2022.

[27] He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In *Proc. the 2017 IEEE International Conference on Computer Vision*, Oct. 2017, pp.2961-2969. DOI: 10.1109/ICCV.2017.322.

[28] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp.580-587. DOI: 10.1109/CVPR.2014.81.

[29] Girshick R. Fast R-CNN. In *Proc. the 2015 IEEE International Conference on Computer Vision*, Dec. 2015, pp.1440-1448. DOI: 10.1109/ICCV.2015.169.

[30] Law H, Deng J. CornerNet: Detecting objects as paired keypoints. In *Proc. the 15th European Conference on Computer Vision*, Sept. 2018, pp.734-750. DOI: 10.1007/978-3-030-01264-9_45.

[31] Newell A, Huang Z, Deng J. Associative embedding: End-to-end learning for joint detection and grouping. In *Proc. the Annual Conference onNeural Information Processing Systems*, Dec. 2017, pp.2277-2287.

[32] Zhou X, Wang D, Krähenbühl P. Objects as points. arXiv:1904.07850, 2019. https://arxiv.org/pdf/1904.07850.pdf, Jan. 2022.

[33] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you need. In *Proc. the Annual Conference on Neural Information Processing Systems*, Dec. 2017, pp.5998-6008.

[34] Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. In *Proc. the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, June 2016, pp.1480-1489. DOI: 10.18653/v1/N16-1174.

[35] Roy A, Saffar M, Vaswani A, Grangier D. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 2021, 9: 53-68. DOI: 10.1162/tacl_a_00353.

[36] Sarlin P E, DeTone D, Malisiewicz T, Rabinovich A. SuperGlue: Learning feature matching with graph neural networks. In *Proc. the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020, pp.4938-4947. DOI: 10.1109/CVPR42600.2020.00499.

[37] Kolesnikov A, Dosovitskiy A, Weissenborn D, Heigold G, Uszkoreit J, Beyer L, Minderer M, Dehghani M, Houlsby N, Gelly S, Unterthiner T, Zhai X. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. the 9th International Conference on Learning Representations*, May 2021.

[38] Wang S, Li B Z, Khabsa M, Fang H, Ma H. Linformer: Self-attention with linear complexity. arXiv:2006.04768, 2020. https://arxiv.org/pdf/2006.04768.pdf, Jan. 2022.

[39] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In *Proc. the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp.7132-7141. DOI: 10.1109/CVPR.2018.00745.

[40] Zhao H, Zhang Y, Liu S, Shi J, Loy C C, Lin D, Jia J. PSANet: Point-wise spatial attention network for scene parsing. In *Proc. the 15th European Conference on Computer Vision*, Sept. 2018, pp.267-283. DOI: 10.1007/978-3-030-01240-3_17.

[41] Wang X, Girshick R, Gupta A, He K. Non-local neural networks. In *Proc. the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp.7794-7803. DOI: 10.1109/CVPR.2018.00813.

[42] Strudel R, Garcia R, Laptev I, Schmid C. Segmenter: Transformer for semantic segmentation. In *Proc. the 2021 IEEE/CVF International Conference on Computer Vision*, Oct. 2021, pp.7262-7272. DOI: 10.1109/ICCV48922.2021.00717.

[43] Wang W, Xie E, Li X, Fan D P, Song K, Liang D, Lu T, Luo P, Shao L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proc. the 2021 IEEE/CVF International Conference on Computer Vision*, Oct. 2021, pp.568-578. DOI: 10.1109/ICCV48922.2021.00061.

[44] Huang Z, Wang X, Wei Y, Huang L, Shi H, Liu W, S Huang T. CCNet: Criss-cross attention for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. DOI: 10.1109/TPAMI.2020.3007032.

[45] Newell A, Yang K, Deng J. Stacked hourglass networks for human pose estimation. In *Proc. the 14th European Conference on Computer Vision*, Oct. 2016, pp.483-499. DOI: 10.1007/978-3-319-46484-8_29.

[46] Lin T Y, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In *Proc. the 2017 IEEE International Conference on Computer Vision*, Oct. 2017, pp.2999-3007. DOI: 10.1109/ICCV.2017.324.

[47] Koziński M, Obozinski G, Marlet R. Beyond procedural facade parsing: Bidirectional alignment via linear programming. In *Proc. the 12th Asian Conference on Computer Vision*, Nov. 2015, pp.79-94. DOI: 10.1007/978-3-319-16817-3_6.

[48] Rahmani K, Huang H, Mayer H. High quality facade segmentation base on structured random forest, region proposal network and rectangular fitting. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2018, IV-2: 223-230. DOI: 10.5194/isprs-annals-IV-2-223-2018.

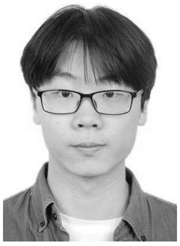
[49] Lin T Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick C L. Microsoft COCO: Common objects in context. In *Proc. the 13th European Conference on Computer Vision*, Sept. 2014, pp.740-755. DOI: 10.1007/978-3-319-10602-1_48.

[50] Zhang Z, Ganesh A, Liang X, Ma Y. TILT: Transform invariant low-rank textures. *International Journal of Computer Vision*, 2012, 99(1): 1-24. DOI: 10.1007/s11263-012-0515-x.

[51] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp.770-778. DOI: 10.1109/CVPR.2016.90.

[52] Yu F, Wang D, Shelhamer E, Darrell T. Deep layer aggregation. In *Proc. the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp.2403-2412. DOI: 10.1109/CVPR.2018.00255.

Yan Tao received his B.S. degree in electronic engineering and information science from University of Science and Technology of China, Hefei, in 2021. Now he is a Master student in University of Science and Technology of China, Hefei. His research interests focus on urban modeling and scene understanding.



Yi-Teng Zhang received his B.S. degree in electronic and information engineering from Zhengzhou University, Zhengzhou, in 2018, and his M.S. degree in signal and information processing from University of Science and Technology of China, Hefei, in 2021. His research interests focus on urban modeling and scene understanding.

Xue-Jin Chen is currently a professor with the National Engineering Laboratory for Brain-Inspired Intelligence Technology and Application, University of Science and Technology of China, Hefei. She received her B.S. degree and Ph.D. degree in electronic circuits and systems from University of Science and Technology of China, Hefei, in 2003 and 2008 respectively. From 2008 to 2010, she conducted research as a postdoctoral scholar in the Department of Computer Science at Yale University, City of New Haven. Her research interests include 3D modeling, geometry processing, and content creation. She has authored or co-authored over 60 papers in these areas. She was one recipient of the Honorable Mention Awards of Computational Visual Media in 2019.