Computer Vision and Image Understanding
journal homepage: www.elsevier.com

RFCNet: Enhancing Urban Segmentation using Regularization, Fusion, and Completion

Xiaowei Zhang**, Daniel Aliaga

Purdue University, West Lafayette, IN, USA

ABSTRACT

Image segmentation is a fundamental task that has benefited from recent advances in machine learning. One type of segmentation, of particular interest to computer vision, is that of urban segmentation. Although recent solutions have leveraged on deep neural networks, approaches usually do not consider regularities appearing in facade structures (e.g., windows are often in groups of similar alignment, size, or spacing patterns) as well as additional urban structures such as building footprints and roofs. Moreover, both satellite and street-view images are often noisy and occluded, thus getting the complete structure segmentation from a partial observation is difficult. Our key observations are that facades and other urban structures exhibit regular structures, and additional views are often available. In this paper, we present a novel framework (**RFCNet**) that consists of three modules to achieve multiple goals. Specifically, we propose **Regularization** to improve the regularities given an initial segmentation, **Fusion** that fuses multiple views of the segmentation, and **Completion** that can infer the complete structure if necessary. Experimental results show that our method outperforms previous state-of-the-art methods quantitatively and qualitatively for multiple facade datasets. Furthermore, by applying our framework to other urban structures (e.g., building footprints and roofs), we demonstrate our approach can be generalized to various pattern types.

1. Introduction

Image segmentation is a fundamental task addressed with machine learning and also plays a crucial role in computer vision. It has many real world applications, including building reconstruction, procedural modeling, and augmented/virtual reality. Recently, deep learning based segmentation has shown its power but two main drawbacks exist: enforcing structured regularities and compensating for missing data (in our case of urban structures). Regarding regularity, Figure 1 (a) shows street-view facades images with windows, doors, and balconies that should be rectangular, horizontally and vertically aligned, spaced equally or with a clear pattern, and/or groups of similar size. Further, this problem is also present in satellite facade images (Figure 1 (b)) and in segmentations of other urban structures such as building footprints and roofs (Figures 1 (c-d)).

Regarding incomplete data, street-view images often have only partial observations of facades and satellite images suffer more due to limitations in resolution, noise, complex camera models, limited viewing angles, and occlusions (Figures 1 (e-f)). In these cases, segmentation of a single image can only show a partial facade structure. Additional work needs to be done in order to complete the facade (e.g., image inpainting/completion). Fortunately, a second or more views for both street-level and satellite are usually available as seen in Figure 1 (g-h). However, only fusing or combining all views still cannot guarantee that the facade is fully covered as also observed in Figure 1 (g-h). Thus, completion techniques are still necessary. Moreover, segmentation of the aforementioned building footprints and roofs may also exhibit incompleteness due to occlusion.

To address the above challenges, our approach takes as input a segmentation from one or more viewpoints as illustrated in Figure 2. Our **Regularization** module improves the regularization of inputs, **Fusion** module fuses multiple views of the urban structure, and **Completion** module infers the complete structure if necessary.

Our framework Regularization, Fusion, and Completion Network (RFCNet) yields improvements over other methods applied to the same data. For several datasets, our method is consistently better than prior work both quantitatively and qualitatively. As far as we know, our work is the first pipeline to handle regularization, fusion, and completion of facades and other urban structures all together using deep learning techniques for both street-view images and satellite-based images. In addition,

**Corresponding author.
e-mail: zhan2597@purdue.edu (Xiaowei Zhang)

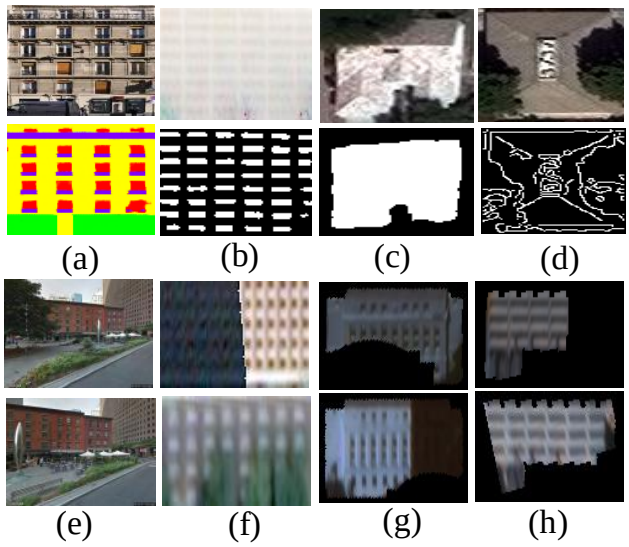

Fig. 1. ***Motivations.*** **(a-b) ECP (Teboul et al., 2011) and satellite facade images, and their segmentations with lack of regular structure and with occlusions. (c-d) SpaceNet (Etten et al., 2018) building images, and their footprint segmentation or Canny (Canny, 1986) edges, showing limited quality. (e) Multi-views of partially occluded Google Street View images. (f) Satellite facades with occlusions/shadows. (g-h) Multi-views of satellite facade images.**

our approach *directly takes the initial segmentation as a starting point and focuses on enhancing the segmentation* without labor-intensive annotation. We hope our work inspires possible future directions for segmentation mask refinement. In short, our main contributions are summarized as follows:

- we propose a novel deep learning based framework (RFC-Net) which improves regularities for patterns, fuses multiple views, and generates plausible and complete urban structures altogether,
- we train our novel pipeline with self-supervision to avoid time-consuming and expensive data annotations,
- we create a synthetic training dataset which incorporates versatile regularities of facade patterns and supports more general facade styles, and perform comprehensive experiments demonstrating usage on both street-view and satellite images, and
- we illustrate usage of our approach for other urban structures such as building footprints and roofs.

2. Related Work

2.1. Segmentation

Semantic segmentation is a classic topic in machine learning and in computer vision. In recent years, with the amazing success of deep learning, many state-of-the-art segmentation networks (Long et al., 2015; Ronneberger et al., 2015; Chen et al., 2015; Badrinarayanan et al., 2017; Chen et al., 2017; Isola et al., 2017; Zhang et al., 2018; Chen et al., 2018; Takikawa et al., 2019) can be applied to urban structures. Specifically, DeepLab (Chen et al., 2015; Chen et al., 2017; Chen et al., 2018) maintains high-resolution by replacing strided convolution with atrous convolution. Encoder-decoder frameworks, like U-Net (Ronneberger et al., 2015), infer high-resolution feature maps by joining the top-down and bottom-up pathways with lateral connections. GAN based frameworks, like Pix2Pix (Isola et al., 2017), consider segmentation as an image-to-image translation problem. However, those approaches most often focus on the general network structure and learning methodology, and include many more content pixels than boundary pixels. This imbalance causes them to produce inaccurate structure edges and cannot ensure structural regularities and completeness (of man-made urban structures).

Nonetheless, there are some proposed deep learning networks particularly focused on urban structures. The DeepFacade approach (Liu et al., 2017) used a fully convolutional network with a special loss function in order to segment facade images. Although their loss function penalized segmented regions that were not horizontally and vertically symmetric, structural regularities were limited to symmetry; fusion and completion was not addressed. FrankenGAN (Kelly et al., 2018) did have a regularizer step to regularize facades and roofs. However, the approach was not data-driven, and only focused on alignment regularity for facade regularization. As for roof regularization, it focuses on roof detail labels (e.g., chimneys, roof windows). The roof itself is assumed to be already regularized. Moreover, none of the aforementioned works dealt with satellite data. Ternausnetv2 (Iglovikov et al., 2018) and Mahmud et al. (2020) are able to perform binary instance segmentation of the building footprints from satellite imagery. Nauata and Furukawa (2020) can vectorize satellite-based buildings and roofs by detecting primitives (e.g., edges). Even so, urban structure regularities and occlusions were not handled in (Iglovikov et al., 2018; Mahmud et al., 2020; Nauata and Furukawa, 2020).

Moreover, some existing works (Zheng et al., 2010; Friedman and Stamos, 2012) try to explore, detect, and use large scale repetitions and regularities of urban structures for consolidating noisy and incomplete scans. However, they focus on 3D data (e.g., point cloud and LiDAR). In addition, there are prior works handling facade occlusions. Kozinski et al. (2015) includes provisions for occlusions but depends on assumed structural priors for object classes and SIFT features. Cohen et al. (2017) also depends on SIFT to extract a set of key-points. On average the satellite facades are only 20x90 pixels in size and thus make it prohibitive to determine such detailed structures.

2.2. Inverse Procedural Modeling

Inverse Procedural Modeling (IPM) attempts to find procedural representations (e.g., rules and/or parameter values), and yields desired and regularized outputs. Nishida et al. (2018) can generate well regularized synthetic building mass and facades by estimating parameters belonging to one of a set of predefined grammars. Nevertheless, it required training a large number of models and performance was limited to the supported building mass and facade grammar styles and it does not work on satellite data. More recently, Zhang et al. (2020) proposed a grammar-based approach that works on satellite data. How-

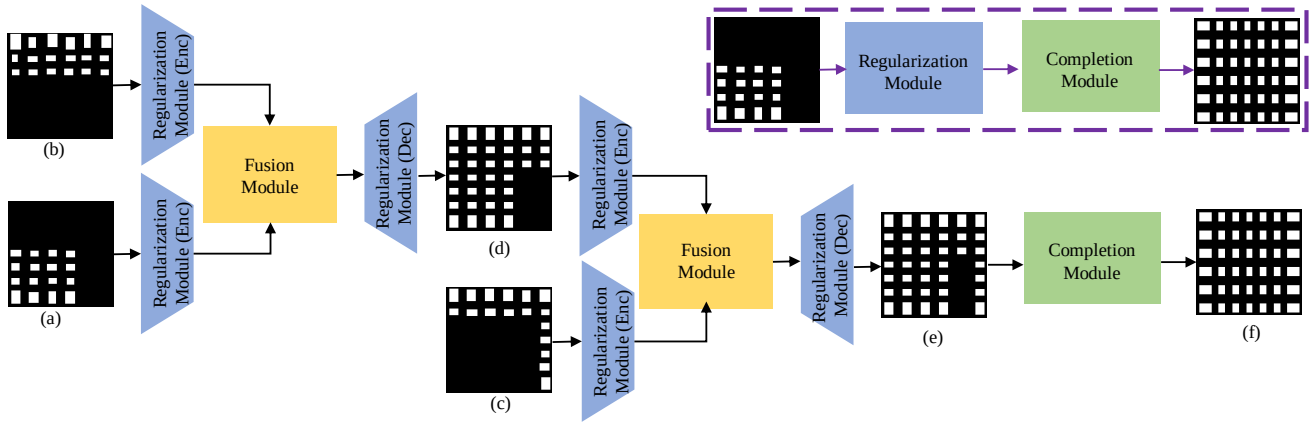

Fig. 2. ***RFCNet.*** **Two scenarios are shown. For single image facade segmentation (dashed box), it will be processed by Regularization and Completion. For multi-view facade segmentation, Fusion will combine the pairwise latent vector of inputs. (a-c) Multi-view facade segmented images. (d-e) Intermediate fused facade images. (f) Final synthetic output of RFCNet.**

ever, it supports only a single facade style and does not address fusion (and completion is only partially addressed).

2.3. Image Fusion

Image fusion merges information from multiple images of the same scene taken from various sensors at different positions and/or different times, hopefully collecting complementary information. However, most recent approaches (e.g., (Prabhakar et al., 2017; Joo et al., 2018; Li et al., 2018; Trinidad et al., 2019; Li et al., 2020; Zhu et al., 2020)) focus on multi-modal fusion (i.e., combining information from different domains). In our case, we are focusing on fusing same-domain images from different viewpoints. Our fusion is more similar to Olszewski et al. (2019) which aggregates multiple volumetric latent representations of the same object, and then applies a simple channel-wise averaging operation to obtain a fused representation. A problem with their basic averaging strategy is that feature maps are fused together without measuring the usefulness of each feature vector; hence, useless and useful features might be mixed together.

2.4. Image Completion

Filling-in missing pixels of an image, often referred to as image completion or image in-painting, is an important task. Deep learning and GAN-based approaches (e.g., (Yu et al., 2018; Zheng et al., 2019; Yu et al., 2019, 2020; Jie Yang, 2020)) have achieved promising results in this task. Compared to real-image completion, segmented-image completion is more challenging due to the lack of color and contextual information. We show comparisons to these approaches in Section 4.

3. Method

In this section, we describe our characterization of pattern regularity and generation of pattern styles for training, present the overall architecture of our RFCNet, and then detail each module of our architecture. Finally, we describe RFCnet implementation.

3.1. Pattern Regularity and Styles

In the following, we describe our assumptions about the pattern structural regularity and styles. We focus on the case of facades to illustrate the details of our method; details for other patterns are defined in an analogous way (see Appendix F for building footprint and roof details).

3.1.1. Structural Regularity

We characterize the structural regularity of facades by the arrangement of their features. *Windows and doors* are the predominant features visible in both street-level and satellite-based facade segmentation. Nonetheless, we also support additional labels for street-level observations (e.g., balconies). The placement of windows/doors can be described by their *alignment* (A), *size* (S) and *spacing* (P) as shown in Figure 3. Since most windows are rectangular and windows shapes are difficult to differentiate in low-resolution satellite images, a window b_i is defined by rectangle $\{x_i, y_i, w_i, h_i\}$ where (x_i, y_i) is its top-left corner and (w_i, h_i) is its size (S). A group of windows is left aligned (A_l) when the x coordinates of the top-left corners are equal. Right (A_r), top (A_t) and bottom (A_b) alignments are defined similarly. The horizontal spacing (P_h) between two horizontally adjacent windows (b_j, b_k) is defined by $x_k - (x_j + w_j)$ (assuming b_k is at the right of b_j). Thus a group of windows has the same horizontal spacing when the computed horizontal spacing among those windows is equal. The vertical spacing (P_v) is defined analogously. Note: We can theoretically grow the space of possible window shapes (e.g., including circular or oval windows) and account for any actual facades. It's listed as future work.

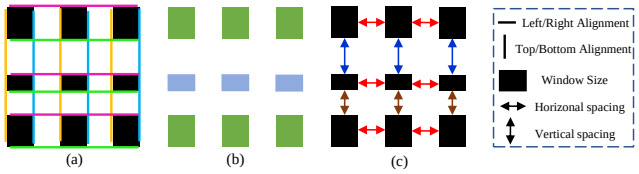

Fig. 3. ***Illustration of Structural Regularity.*** **(a) Left, right, top and bottom alignments are in different colors. (b) Different window sizes are in different colors. (c) Horizontal and vertical spacing are in different colors.**

3.1.2. Style Generation

Within a facade there can be one or more groups of windows/doors exhibiting different combinations of the aforementioned structural characteristics $A_l/A_r/A_t/A_b$, S, and P_h/P_v. A particular combination of characteristics defines a *facade style*. For example, the facade style (a) in Figure 4 is based on the combination of constraints: A_l and A_r for each column of windows, A_t and A_b for each floor, windows of same size S, and same P_h and P_v spacing. (b) in the same figure differs from (a) by having more than one group of windows of the same S. The facade style (c) differs from (a) by having 2 groups of P_h in the facade.

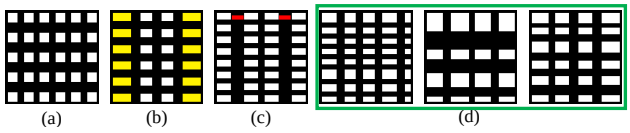

Fig. 4. ***Example Facade Styles.*** **(a-d) show progressively more general facade styles, with (d) being supported by our approach.**

In the specific case of facade segmentation, prior work assumes specific structural constraints. For instance, Zhang et al. (2020) addresses facade segmentation and completion but only for facades of style (a). Nishida et al. (2018) defines a specific set of 16 synthetic facade styles, similar to styles (a-c). For each proposed facade style, they generated enough synthetic training images to train a set of deep networks. For published facade datasets like CMP (Tyleček and Šára, 2013) or ECP (Teboul et al., 2011), the supported facades are limited to the styles observed in the cities where the facades were collected. Further, they assume facades are captured at sufficiently high resolution and completeness to observe the supported stylistic details. Training based on those datasets will fail when handling other facade styles or satellite facade images.

Unlike the aforementioned methods, our approach is based on a much more general set of assumptions. For facades, our approach works as long as the facade satisfies the near minimal constraints of basic alignment (i.e., A_l and A_r for groups of columns of windows, A_t and A_b for groups of floors as seen in real facade images from Figure 1), and groups with same size S and groups with similar spacing P_h and/or P_v. During training, we will generate a very large number of training examples that essentially will include all the prior sets of specific styles as well as other more general facade styles like (d) of Figure 4. To show the facade generality of our set of assumptions, we test several models against the styles in Figure 4. In Appendix D, we summarizes the quality of the results and clearly show that our method works comparatively best.

3.2. Architecture

Our RFCNet, as illustrated in Figure 2, contains three sequential modules: Regularization module (R), Fusion module (F), and Completion module (C). RFCNet takes as input a segmentation from one or more viewpoints. For a single viewpoint, the input segmentation passes through Regularization and Completion. For the multi-view scenario, inputs will be successively combined by Fusion in pairs. In both cases, the output is a well-regularized, crisp and complete synthetic structure. Moreover, the whole network is trained in an end-to-end manner.

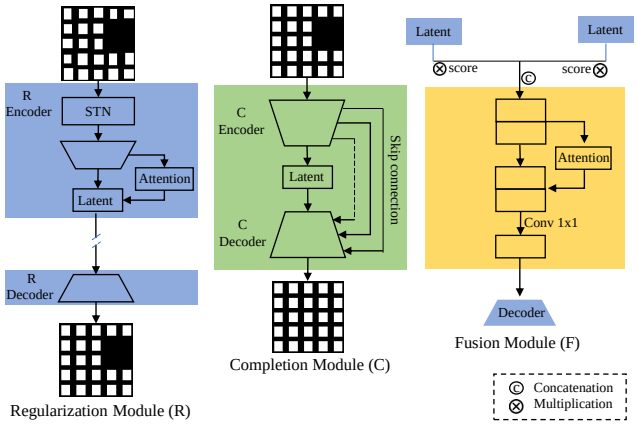

Fig. 5. ***Modules.*** **Structural details of Regularization, Completion and Fusion modules.**

3.2.1. Regularization

The design of our Regularization module is mainly based on a Convolutional Autoencoder (Masci et al., 2011). Our Regularization consists of two main blocks as shown at the left of Figure 5: an encoder part E^R and a decoder part D^R. To be specific, E^R takes the raw (un-regularized) segmentation $I^R \in \mathbb{R}^{H\times W\times C}$ as input, and first passes through a spatial transform network (STN) (Jaderberg et al., 2015) which predicts a global affine transformation T to align the facade I^R to $I^R_t = T(I^R)$. I^R_t will be downsampled by a series of 2D convolutions layers into a lower dimensional latent representation $Z^R \in \mathbb{R}^{H'\times W'\times C'}$ that contains the informative content of the facade. D^R is trained to up-sample and reconstruct a regularized and synthetic output $\tilde{I}^R \in \mathbb{R}^{H\times W\times C}$ from Z^R. Each layer is either a convolutional layer (by default) or a residual block (He et al., 2016). We also add an attention module CBAM (Woo et al., 2018) to bias allocation of the most informative feature expressions and simultaneously suppress the less useful ones. Please see the attention analysis in Appendix A.

3.2.2. Fusion

In order to fuse multiple facade segmentations $\{..., I^R_j, I^R_k, ...\}$, our Fusion module takes a pair of encoded latent representations $(Z^R_j, Z^R_k) \in \mathbb{R}^{H'\times W'\times C'}$ from E^R as inputs each time and generates an accumulated representation Z^F_{jk}. Then Z^F_{jk} passes through D^R for subsequent processing. The Fusion module is shown at the right of Figure 5. Our Fusion naturally extends to an arbitrary number of inputs in this way (e.g., an example of 3 inputs shown in Figure 2). Unlike the work Olszewski et al. (2019) using basic averaging-based fusion (or, max, min, or addition), our proposed Fusion not only incorporates all those basic fusion strategies, but also learns the weights of how to fuse. In the beginning, (Z^R_j, Z^R_k) are scaled by their confidence values (or score) (w^F_j, w^F_k), and then they are concatenated together to form a deeper representation $Z^F_c = concat(w^F_j \times Z^R_j, w^F_k \times Z^R_k)$

and $Z_c^F \in \mathbb{R}^{H' \times W' \times 2C'}$. The confidence value corresponds to the quality of the input facade (e.g., given different views of the same facade, the user may provide one of 1.0, 0.75 or 0.5 that correspond to high, medium and low quality images respectively). This confidence value especially helps when we use satellite based segmentation as illustrated in Appendix B. Next, the concatenated representation Z_c^F is compressed by passing through a 1×1 convolutional layer to reduce the depth of channels to the original size C'. An attention module CBAM is also added to measure the usefulness of each feature vector, and filter out less important features.

3.2.3. Completion

Our Completion module takes the partially viewed and well-regularized segmentation $I^C \in \mathbb{R}^{H \times W \times C}$ as input, and then generates a well-regularized and complete synthetic output $\tilde{I}^C \in \mathbb{R}^{H \times W \times C}$ as shown in the middle of Figure 5. The Completion architecture is similar to Regularization but with several notable differences. STN is not necessary since the input has been aligned during Regularization. Since completion is more about propagating feature information globally, the bottleneck latent representation is fully connected to the previous layer. Skip connections (Ronneberger et al., 2015) are added between the layer in the encoder and the corresponding layer in the decoder. Please see Appendix C for how a fully connected layer improves completion. Note: Since our Fusion module fuses a pair of encoded latent representations of two views of the same facades, training a combined Regularization-Completion module does not guarantee that the latent code only represents the visible part of each façade – therefore the Fusion process would work incorrectly.

3.3. Implementation

We perform self-supervised training using synthetic data for our proposed network implemented in PyTorch (Paszke et al., 2017). The weights are trained by the Adam (Kingma and Ba, 2015) optimizer where initial learning rate is set to 1e-3. Our typical input image sizes are $(H, W, C) = (128, 128, 1)$ and latent space dimensions are $(H', W', C') = (4, 4, 256)$. It runs on an Intel i9 workstation with NVIDIA RTX 2080 8GB cards. In the following, we describe training, loss function, and testing.

We generate 500,000 synthetic images for RFCNet training as per our general set of style assumptions in Section 3.1.2. For data augmentation, we first apply a random occlusion mask M to create a masked image $\hat{I} = I \odot (1 - M)$. Then, to add noise and irregularities to facade images, we apply random local window deformations T_{local} (e.g., translation and scaling) and global affine transformations T_{global} (e.g., translation and rotation) yielding the final input training image $\hat{I}_t = T_{global}(T_{local}(\hat{I}))$. Likewise, in terms of data transformation for other patterns, please find details in Appendix F. Thus our aforementioned $I^R = \hat{I}_t$. To train Regularization and Completion end-to-end, we also need that the output of Regularization goes into Completion directly, thus $I^C = \tilde{I}^R$.

We formulate Regularization as a supervised learning problem and compute its loss $\mathcal{L}^R$ as the weighted sum of the squared $L2$ losses of the image and its spatial gradient map (using Sobel operator). It is defined as:

$$\begin{aligned} \mathcal{L}^R &= \mathcal{L}_r^R + \lambda \mathcal{L}_s^R, \\ \mathcal{L}_r^R &= \|\tilde{I}^R - \hat{I}\|_2^2, \\ \mathcal{L}_s^R &= \|f(\tilde{I}^R) - f(\hat{I})\|_2^2, \end{aligned} \tag{1}$$

where f stands for generating the spatial gradient map (with an empirically determined $\lambda = 10$). In a similar way, the loss function for Completion $\mathcal{L}^C$ is defined as follows (with $\beta = 10$ determined empirically to work well):

$$\begin{aligned} \mathcal{L}^C &= \mathcal{L}_r^C + \beta \mathcal{L}_s^C, \\ \mathcal{L}_r^C &= \|\tilde{I}^C - I\|_2^2, \\ \mathcal{L}_s^C &= \|f(\tilde{I}^C) - f(I))\|_2^2. \end{aligned} \tag{2}$$

Thus the total loss function is the following:

$$\mathcal{L} = \mathcal{L}^R + \mathcal{L}^C. \tag{3}$$

After we have trained Regularization and Completion, we use the sub-parts E^R and D_R to help train Fusion. Since the input for the Fusion module is a pair of encoded feature representations (Z_j^R, Z_k^R) generated by E^R and the output goes into D_R, we freeze E^R and D^R and then only update the Fusion block during its training with a fusion loss function $\mathcal{L}^F$ defined in a similar way as $\mathcal{L}^R$. In order to generate (Z_j^R, Z_k^R) for a given I, two masks, local deformations and global transformations (e.g., two different views of the same facade) are needed to get a pair of (I_j^R, I_k^R).

4. Experiments

We quantitatively and qualitatively evaluate our approach on building facades, for which we have ample ground truth. We also show preliminary qualitative results for building footprints and roofs.

4.1. Facade Datasets

For facades, we test on three datasets: ECP dataset (Teboul et al., 2011), a WorldView3 Satellite dataset (WVS), and a Google Street View dataset (GSV). Our method supports multiple facade labels and we show such results in Figure 9 (a) and more in Appendix F. However, we focus on window patterns in all datasets for consistence since they are most important and obvious to have regularities, and other labels/details are difficult to see from satellite imagery.

4.1.1. ECP

ECP dataset consists of 104 images of building facades. All images in ECP dataset contain rectified and complete building facades in Paris. We use the annotations provided by Mathias et al. (2016). To show that our approach can enhance the initial facade segmentation from a variety of models, we train and test three segmentation models whose architectures are significantly different: U-Net (Ronneberger et al., 2015), Pix2Pix (Isola et al., 2017), and DeepLabv3+ (Chen et al., 2018). We leave 20 images for evaluation (4 of the 20 images are manually masked for completion evaluation).

4.1.2. WVS

We present a dataset of satellite-captured facades, and it includes 152 rectified high-resolution satellite images (i.e., 0.3m per pixel) from WorldView3. The images have been manually annotated with two labels: one for windows/doors and the other for the walls. These images contain complete facades, partially occluded facades (e.g., due to shadows, trees, etc.), and multi-view facades. We train and test three segmentation models: Pix2Pix (Isola et al., 2017), EncNet (Zhang et al., 2018) and DeepLabv3+ (Chen et al., 2018). We leave 20 complete, 4 partially occluded and 4 sets of multi-view facades for evaluation.

4.1.3. GSV

We collected 4 complete facades, 4 occluded facades and 4 sets of multi-view facades from Google Maps. We rectified the images and manually created annotations. For segmentation, we directly use the models trained for ECP.

4.2. Facade Evaluation Metrics

We rigorously evaluate our RFCNet in the case of facade segmentation. In particular, we evaluate two ways: facade correctness and facade regularization. Facade correctness focuses on the pixel-level performance of the facade. Facade regularization measures the regularities of the facade. Please see the evaluation code in our supplementary materials.

4.2.1. Correctness

For facade correctness evaluation, we use the following statistical measures:

- **Accuracy:** $\frac{TP+TN}{ALL}$
- **Precision:** $\frac{TP}{TP+FP}$
- **Recall:** $\frac{TP}{TP+FN}$
- **F1:** $2 * \frac{Precision*Recall}{Precision+Recall}$

with true positives TP, false positives FP, true negatives TN, and false negatives FN for window/door class.

4.2.2. Regularization

Three metric error terms are defined to measure the regularization of a facade layout: alignment (E_a), size (E_s) and spacing (E_p). The errors measure the deviation from having groups of perfect alignment, groups of equal size, and groups of equal spacing. We adapt and modify the relevant definitions of Jiang et al. (2016).

- **Group:** We use a threshold t to split the window elements into a set of groups $G = \{..., g_i, ...\}$ for the regularization error terms. A candidate group $g_i = (E_i, V_i)$ contains a set of window elements E_i that share the regularization term, and a set of values V_i (e.g., it is a set of x coordinate of left-corner from E_i for left alignment) that will be used to compute the corresponding regularization error. Please refer to Section 3.1.1 for definitions of these values.
- **Alignment Error:** E_a is defined as:

$$E_a = \sum_{A_i}(\sum_{g_j} \frac{stdvar(g_j)}{scale(g_j)} + w_a\|G\|), \tag{4}$$

where A_i stands for one alignment type among top, bottom, left and right alignments. g_j is a candidate group of A_i. $stdvar(g_i)$ measures the standard deviation of V_i in g_i. $scale(g_i)$ is used to scale the error. For left and right alignment, it is equal to the minimal width of window elements E_i in g_i. For top and bottom alignment, it is equal to the minimal height of E_i. $\|G\|$ is the number of candidate groups of A_i. We add $\|G\|$ to encourage fewer and larger groups. w_a is a weight that balances the two terms (e.g., $w_a = 0.01$).

- **Size Error:** E_s is defined as:

$$E_s = \sum_{g_j} \frac{stdvar(g_j)}{scale(g_j)} + w_s\|G\|, \tag{5}$$

where g_j is a candidate group that has the same window size. $stdvar(g_i)$ measures the standard deviation of V_i in g_i. $scale(g_i)$ is equal to minimal height or width of E_i in g_i. $\|G\|$ is the number of groups. w_s is a weight that balances the two terms (e.g., $w_s = 0.01$).

- **Spacing Error:** E_p is defined as:

$$E_p = \sum_{P_i}(\sum_{g_j} \frac{stdvar(g_j)}{scale(g_j)} + w_p\|G\|), \tag{6}$$

where P_i stands for horizontal or vertical spacing. g_j is a candidate group of P_i. $stdvar(g_i)$ measures the standard deviation of V_i in g_i. $scale(g_i)$ is equal to minimal spacing of V_i in g_i. $\|G\|$ is the number of candidate groups of P_i. w_p is a weight that balances the two terms (i.e., we usually set $w_p = 0.01$).

4.3. Facade Comparison

We evaluate RFCNet on the aforementioned ECP, WVS and GSV datasets with our defined evaluation metrics. To verify effectiveness of our method on enhancing initial segmentation, we compared our method to three segmentation models, state-of-the-art image completion models, and IPM methods both qualitatively and quantitatively. The comparison exhibits our approach shows a significant improvement. In the following sections, we only show comparison with one initial segmentation model. Please see Appendix E for more comparisons.

4.3.1. Regularization

We apply our Regularization to initial segmentations for each of our three datasets and in all cases achieve better performance. The initial segmentation for each of the datasets is the first row in each group of Table 1 (e.g., for WVS, we improve the accuracy of the initial segmentation of Pix2Pix by **6.2%** and reduce alignment error E_a by **60.8%**). In addition, we retrain models in IPM methods (Nishida et al. (2018); Zhang et al. (2020)) using corresponding datasets for fair comparison. It shows our

method improves facade correctness compared to them (e.g., accuracy is improved by **30.2%** for ECP, **30.1%** for GSV, and **4%** for WVS). Moreover, as shown in Figure 6, our module generates visually pleasant facade structures.

Table 1. ***Regularization Quantitative Comparison.*** **We compare our Regularization (R) with the initial facade segmentation and IPM methods for ECP, WVS and GSV datasets. For facade correctness, higher is better. For facade regularization error, lower is better. Note: IPM methods generate regularized outputs, so regularization error is close to 0 but correctness is lower than others.**

Dataset	Method	Facade Correctness				Facade Regulariz. Error		
		Acc.	Pre.	Rec.	F1	E_a	E_s	E_p
ECP	DeepLabv3+	96.4%	84.7%	94.6%	89.2%	1.01	0.05	0.18
	Nishida et al.	69.0%	50.8%	53.3%	51.7%	—	—	—
	R	**99.2%**	**95.3%**	**99.4%**	**97.2%**	**0.32**	**0.04**	**0.12**
WVS	Pix2Pix	87.2%	70.1%	91.9%	79.0%	0.74	0.12	0.27
	Zhang et al.	89.4%	80.5%	84.6%	82.2%	—	—	—
	R	**93.4%**	**81.6%**	**96.8%**	**88.2%**	**0.29**	**0.08**	**0.16**
GSV	U-Net	90.5%	75.0%	90.0%	81.6%	0.76	0.05	0.22
	Nishida et al.	68.0%	50.2%	51.6%	50.7%	—	—	—
	R	**98.1%**	**92.8%**	**99.5%**	**95.9%**	**0.25**	**0.04**	**0.13**

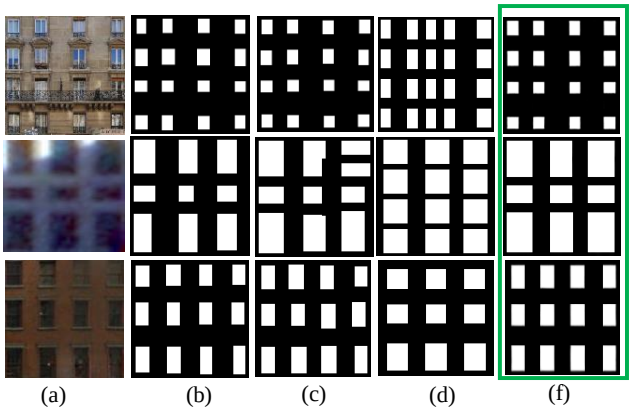

Fig. 6. ***Regularization Qualitative Comparison.*** **(a) Facade images from ECP, WVS and GSV respectively. (b) Ground Truth. (c) Initial Segmentation. (d) IPM results. (f) Our Regularization.**

4.3.2. Regularization and Completion

For incomplete or partially occluded facade segmentation, as shown in Table 2, our Regularization and Completion (R & C) not only significantly improves the facade correctness and regularization metrics (e.g., for WVS, we improve the accuracy of the initial segmentation of Pix2Pix by **8.4%** and reduce alignment error E_a by **72.4%**), but also achieves better performance compared to the initial segmentation augmented by the image in-painting method DeepFill (Yu et al., 2018) (e.g., accuracy improved by **5.4%** and alignment error E_a reduced by **74.7%** for WVS). For a fair comparison, we refine DeepFill using our facade synthetic dataset. In addition, our method improves IPM methods with respect to facade correctness (e.g., accuracy improved by **29.5%** for ECP, **26.5%** for GSV, and **3.1%** for WVS). Further, our R & C outperforms the single Regularization module in terms of accuracy and recall. As illustrated in Figure 7, the outputs of our Regularization and Completion are visually appealing as well.

Table 2. ***Regularization and Completion Quantitative Comparison.*** **After applying Regularization and Completion (R & C) to the initial segmentation, we compare our results to the segmentation, segmentation after completion using DeepFill, IPM methods and our Regularization (R). Note: DeepFill means DeepFill takes the initial segmentation of the initial segmentation (i.e., Pix2Pix) as inputs.**

Dataset	Method	Facade Correctness				Facade Regulariz. Error		
		Acc.	Pre.	Rec.	F1	E_a	E_s	E_p
ECP	DeepLabv3+	91.0%	74.0%	74.8%	74.1%	0.77	0.05	0.18
	DeepFill	93.2%	73.5%	95.8%	82.6%	0.80	0.04	0.21
	Nishida et al.	67.4%	50.4%	54.4%	52.3%	—	—	—
	R	95.3%	**92.8%**	78.4%	86.6%	0.27	0.03	**0.12**
	R & C	**96.9%**	84.6%	**99.7%**	**91.7%**	**0.24**	**0.03**	0.15
WVS	Pix2Pix	84.8%	77.2%	65.9%	71.1%	0.87	0.07	0.32
	DeepFill	87.8%	75.5%	84.1%	79.5%	0.95	0.07	0.33
	Zhang et al.	90.1%	82.3%	87.7%	84.5%	—	—	—
	R	89.9%	**88.3%**	74.3%	80.9%	**0.21**	0.05	**0.14**
	R & C	**93.2%**	83.2%	**94.7%**	**88.5%**	0.24	**0.05**	0.19
GSV	U-Net	82.7%	74.7%	55.3%	60.9%	0.71	0.06	0.14
	DeepFill	87.3%	75.5%	77.5%	75.7%	0.89	0.07	0.26
	Nishida et al.	67.1%	50.1%	51.3%	50.5%	—	—	—
	R	88.5%	**91.6%**	65.4%	78.3%	**0.20**	**0.04**	**0.07**
	R & C	**93.6%**	83.8%	**93.4%**	**88.0%**	0.26	0.06	0.12

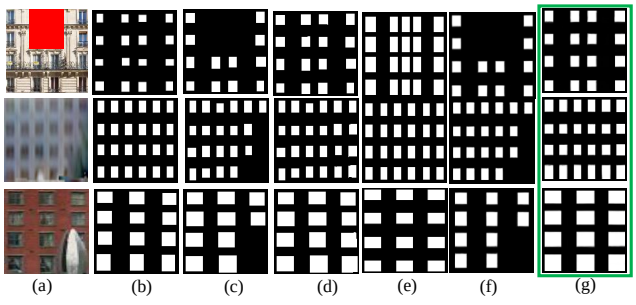

Fig. 7. ***Regularization and Completion Qualitative Comparison.*** **(a) Occluded facade images from ECP, WVS and GSV respectively. (b) Ground Truth. (c) Initial Segmentation. (d) Segmentation completed by DeepFill. (e) IPM results. (f) Our Regularization. (g) Our Regularization and Completion. Note: We manually mask ECP facade images shown in red box.**

4.3.3. RFCNet

For partially-occluded facade segmentation with additional views, as shown in Table 3, our RFCNet achieves better facade results when evaluated for facade correctness and regularization as compared to the segmentation of the first view (e.g., for WVS, improves the segmentation accuracy by **9.1%** and reduces alignment error E_a by **60.8%**) and the segmentation completed using DeepFill (e.g., for WVS, improves accuracy by **4.4%** and reduces E_a by **70.1%**). Moreover, our RFCNet obtains better performance compared with only applying our Regularization and Completion to the first view. In addition, our method improves facade correctness compared to IPM methods (e.g., accuracy improved by **28.4%** for GSV and **5.6%** for WVS). What's more, our RFCNet improves accuracy and recall compared to both R & C and R. As demonstrated in Figure 8, our RFCNet results are qualitatively preferable.

Table 3. *RFCNet Quantitative Comparison.* We compare the initial facade segmentation, the segmentation completed by DeepFill, IPM methods, and the outputs after applying our Regularization (R) and our R & C to the segmentation for the first view in WVS and GSV. Further, we evaluate the output after fusing additional views by applying our whole RFCNet.

Dataset	Method	Facade Correctness				Facade Regulariz. Error		
		Acc.	Pre.	Rec.	F1	E_a	E_s	E_p
WVS	Pix2Pix	85.9%	86.1%	65.8%	74.0%	0.51	0.09	0.27
	DeepFill	90.6%	85.1%	85.2%	85.1%	0.67	0.13	0.43
	Zhang et al.	89.4%	80.7%	86.2%	83.3%	—	—	—
	R	90.1%	**94.7%**	72.8%	81.9%	**0.19**	**0.06**	**0.14**
	R & C	93.5%	85.5%	95.1%	90.0%	0.20	0.10	0.15
	RFC	**95.0%**	88.7%	**96.0%**	**92.2%**	0.20	0.09	0.15
GSV	U-Net	83.6%	80.0%	53.8%	62.2%	0.71	0.07	0.15
	DeepFill	87.2%	79.2%	71.1%	74.0%	0.89	0.11	0.19
	Nishida et al.	66.9%	50.3%	50.7%	50.4%	—	—	—
	R	89.7%	**94.0%**	67.6%	80.5%	0.20	0.06	0.11
	R & C	91.6%	81.3%	88.1%	83.9%	0.28	**0.06**	0.14
	RFC	**95.3%**	86.4%	**95.2%**	**90.4%**	**0.15**	0.06	**0.09**

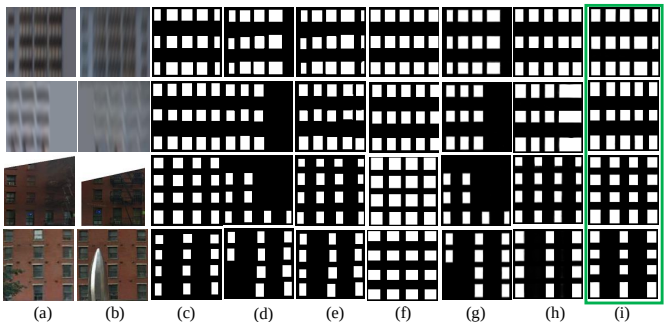

Fig. 8. *RFCNet Qualitative Comparison.* (a) First view of facade images from WVS and GSV. (b) Additional views. (c) Ground Truth. (d) Segmentation of the first view. (e) Segmentation completed by DeepFill. (f) IPM results. (g) Our Regularization. (h) Our R & C. (i) Our entire RFCNet.

4.4. Footprints and Roofs

We evaluate our approach on SpaceNet dataset (Etten et al., 2018) for footprints and roofs. The dataset includes high-resolution (0.3m per pixel) satellite imagery from several cities. Building footprint annotations are already available and we manually annotate the roof structures. In addition, we train Mask R-CNN (He et al., 2017) to get the initial footprint segmentation and we take Canny edges (Canny, 1986) as inputs for roof pattern.

As shown in Figure 9 (b), our method improves the regularities of footprints (e.g., straight walls, parallel walls and corners with right angles) compared to the initial segmentation. As illustrated in Figure 9 (c), the roof outputs of our approach are visually pleasant as well (e.g., roofs with rectangular components, ridges parallel to rectangle edges, hips perfectly connected to the ridges). We provide more examples and details (e.g., assumptions of structural regularities and styles, and synthetic data transformation) in Appendix F.

4.5. Failure examples

Although we support a very wide range of facade styles, there are always exceptions (e.g., columns of windows that are purposefully unaligned). Currently for styles outside our assumptions, our approach gives its best guess. In the first example

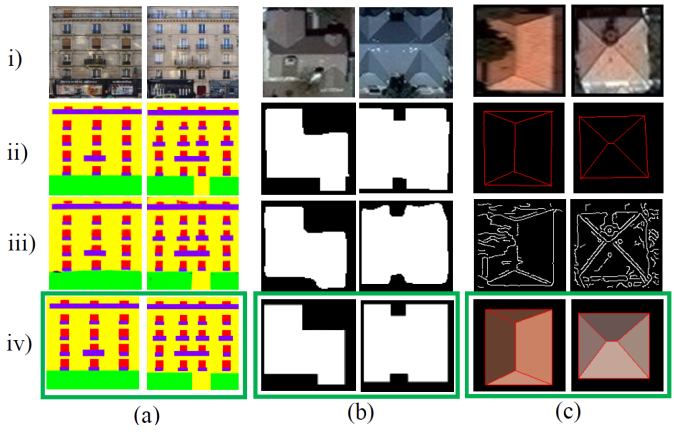

Fig. 9. *Additional Urban Structures.* We show (a) ECP facades with multiple labels, (b) building footprints, and (c) roofs. For each, i) ECP or SpaceNet provided image, ii) ground truth, iii) initial segmentation or Canny edges, and iv) our RFCNet result. Note: More results are in Technical Appendix Section F. Roof outputs are rendered using the average color.

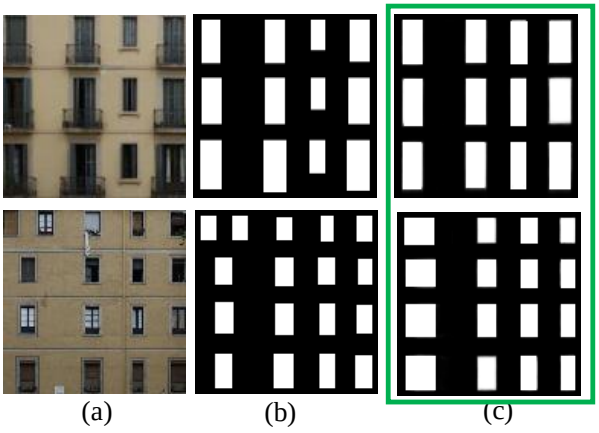

Fig. 10. *Failure Examples.* (a) Facade images. (b) Initial segmentation. (c) Our results.

of Figure 10, due to our A_b (bottom alignment) assumption for each floor, our method enforces the facade output to satisfy A_b regularity. As for the second one, there are six columns in the facade image. However, our approach generates four columns of windows by combining the first three columns into one column with wider windows. We explain the reasons behind the missing columns. Figure 11 (a) shows the original segmentation and our result of the second example in Figure 10. As you can see, the two colorful (blue and orange) line segments cross the window below , and it causes no spacing among these columns of windows in the segmentation image which is outside of our facade assumptions. Moreover, we manually remove parts of the top two windows to leave spacing for each column, shown in Figure 11 (b). We generate reasonable output this time.

In theory, our framework can handle these failure scenarios by adding more versatile facade patterns to our synthetic training datasets. This is listed as future work.

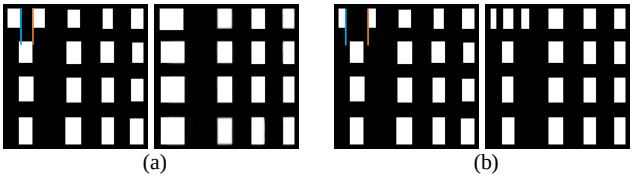

Fig. 11. ***More about Failure Examples.*** **(a) Original segmentation and our results. (b) Manually modified segmentation and our results.**

5. Conclusion

We have proposed a novel deep learning based framework RFCNet, which improves regularities of urban structures, fuses multiple views, and generates plausible and complete structures. Through comprehensive experiments, we show our approach significantly enhances urban segmentation for multiple datasets. However, our approach has some limitations. Although we support a very wide range of styles, there are always exceptions. Please find visual failure examples in Section 4.5.

Our approach has several avenues of future work. For example, we would like to generate geometric structures based on our refined segmentation. Also, we would like to combine our framework with the segmentation model so as to handle real images directly. Moreover, we would like to support more window shapes (e.g., circular windows, oval windows, etc.). Finally, we are also interested in applying our framework to other pattern-like applications (e.g., textures, sketches, floor-plans, etc.).

Acknowledgements

This research was funded in part by National Science Foundation grants #1816514 *CHS: Small: Functional Proceduralization of 3D Geometric Models*, #1835739 *U-Cube: A Cyberinfrastructure for Unified and Ubiquitous Urban Canopy Parameterization*, and #2107096 *Deep Generative Modeling for Urban and Archaeological Recovery*.

Supplementary Material

Supplementary material is provided along with the manuscript, and includes the evaluation code of our three facade datasets.

Appendix A. Attention

In Table A.1, we show an analysis of our network without an attention block, with SEA (Hu et al., 2018), with CBAM (Woo et al., 2018) or with DA (Fu et al., 2019). It shows using CBAM improves our network the most.

Appendix B. Confidence Value

We analyze how confidence values help to improve the performance and robustness of our approach in Figure B.1. Even fused with a noisy and low quality view of a facade, our method is still able to retrieve useful information and generate a good output.

Table A.1. ***Attention Analysis.*** **We employ different attention blocks and evaluate them on different facade styles as shown in Figure 4 of the paper in terms of $\mathcal{L}^R$ defined in Equation 1 of the paper.**

Models	(a)	(b)	(c)	(d)
No Attention	0.0068	0.0074	0.0075	0.0086
SEA	**0.0061**	0.0070	0.0071	0.0082
CBAM	**0.0061**	**0.0069**	**0.0069**	**0.0081**
DA	0.065	0.0072	0.0071	0.0084

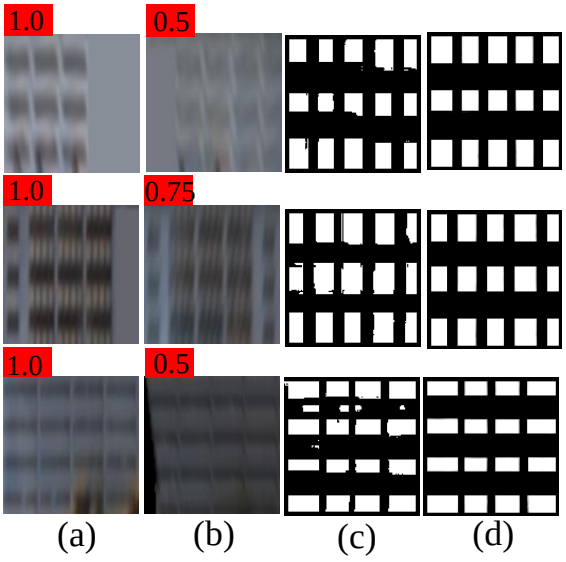

Fig. B.1. ***Confidence Value.*** **(a) and (b) are two views of satellite facade examples. Confidence values are inside the red box. (c) Our outputs without applying confidence values. (d) Our outputs with confidence values.**

Appendix C. Latent Vector

In Table C.1, we conduct an analysis of our Completion module using a convolutional latent vector, or a fully-connected latent vector. It shows the fully-connected latent vector has better performance.

Table C.1. ***Latent Vector Analysis.*** **We experiment on different latent vector types for our Completion module and evaluate them on different facade styles as shown in Figure 4 of the paper in terms of $\mathcal{L}^C$ defined in Equation 2 of the paper.**

Latent Vectors	(a)	(b)	(c)	(d)
Convolutional	0.0055	0.0066	0.0064	0.0088
Fully-connected	**0.0045**	**0.0058**	**0.0055**	**0.0076**

Appendix D. Facade Generality

To show the generality of our facade set of assumptions, we test several models against the styles in Figure 4 of our paper. Model *I* is trained using only facades of style *a* and Model *II* is trained using facades of style *c*. Both models, and our RFCNet

are tested against 1000 images of each of the styles a, b, c, and d. Table D.1 summarizes the quality of the results and clearly shows that RFCNet works best.

Table D.1. ***Evaluation on different models.*** **We evaluate styles (a-d) from Figure 4 of the paper on Model *I*, Model *II* and RFCNet in terms of $\mathcal{L}$ defined in Equation 3 of the paper.**

Styles	Model *I*	Model *II*	RFCNet
(a)	0.0024	0.0290	0.0096
(b)	0.0800	0.1062	0.0114
(c)	0.0530	0.0052	0.0118
(d)	0.1164	0.0784	0.0150
Average	0.0630	0.0547	**0.0120**

Appendix E. Comparison

We evaluate RFCNet on the aforementioned ECP, WVS and GSV datasets with our defined evaluation metrics. We show comparisons with other initial segmentation models, state-of-the-art image completion models and IPM methods both qualitatively and quantitatively in the following sections. The comparison shows our approach shows a significant improvement.

Appendix E.1. Regularization

Regarding complete facade, our Regularization (R) achieves better performance across our three datasets with respect to facade correctness and regularization metrics compared with initial segmentation models: U-Net (Ronneberger et al., 2015), Pix2Pix (Isola et al., 2017), EncNet (Zhang et al., 2018) and DeepLabv3+ (Chen et al., 2018) as shown in Table E.1.

Table E.1. ***Regularization Quantitative Comparison.*** **We compare the initial facade segmentation to the output after applying our Regularization (R) for ECP, WVS and GSV datasets.**

Dataset	Method	Facade Correctness				Facade Regulariz. Error		
		Acc.	Pre.	Rec.	F1	E_a	E_s	E_p
ECP	U-Net	96.6%	87.8%	92.1%	89.7%	0.98	0.60	0.17
	R	**98.9%**	**95.9%**	**97.7%**	**96.8%**	**0.33**	**0.04**	**0.11**
ECP	Pix2Pix	95.4%	89.2%	93.2%	90.9%	0.90	0.05	0.21
	R	**99.4%**	**98.2%**	**99.1%**	**98.6%**	**0.32**	**0.04**	**0.15**
WVS	EncNet	87.2%	83.9%	77.3%	79.4%	0.87	0.12	0.18
	R	**93.3%**	**91.4%**	**87.8%**	**89.1%**	**0.32**	**0.07**	**0.12**
WVS	DeepLabv3+	86.1%	76.5%	82.9%	78.5%	0.89	0.12	0.15
	R	**94.0%**	**88.2%**	**92.9%**	**90.0%**	**0.31**	**0.08**	**0.13**
GSV	Pix2Pix	92.6%	84.8%	88.1%	86.2%	0.68	0.07	0.20
	R	**99.1%**	**96.6%**	**98.7%**	**97.6%**	**0.19**	**0.06**	**0.13**
GSV	DeepLabv3+	91.4%	80.6%	87.9%	83.0%	0.76	0.08	0.28
	R	**98.4%**	**96.6%**	**97.0%**	**96.7%**	**0.26**	**0.06**	**0.20**

Appendix E.2. Regularization and Completion

For incomplete or partially occluded facade segmentation, as shown in Table E.2, our Regularization and Completion (R & C) not only significantly improves the facade correctness and regularization metrics, but also achieves better performance compared to various initial segmentations augmented by the image in-painting method DeepFill (Yu et al., 2018).

Table E.2. ***Regularization and Completion Quantitative Comparison.*** **We compare the initial facade segmentation to the initial segmentation after completion using DeepFill, and to after applying our Regularization and Completion (R & C) for all three datasets.**

Dataset	Method	Facade Correctness				Facade Regulariz. Error		
		Acc.	Pre.	Rec.	F1	E_a	E_s	E_p
ECP	U-Net	90.8%	75.1%	73.6%	74.3%	0.57	0.05	0.18
	DeepFill	93.3%	74.6%	95.0%	83.4%	0.65	**0.04**	0.20
	R & C	**97.5%**	**87.4%**	**99.7%**	**93.1%**	**0.24**	0.04	**0.12**
ECP	Pix2Pix	89.7%	90.6%	72.8%	80.7%	0.80	0.05	0.17
	DeepFill	94.9%	90.2%	92.6%	91.4%	0.84	0.05	0.23
	R & C	**97.9%**	**93.2%**	**99.3%**	**96.1%**	**0.28**	**0.04**	**0.14**
WVS	EncNet	84.6%	84.4%	65.3%	73.3%	0.88	0.07	0.21
	DeepFill	86.8%	81.8%	76.6%	79.0%	0.97	0.07	0.33
	R & C	**92.8%**	**86.1%**	**93.2%**	**89.3%**	**0.23**	**0.05**	**0.15**
WVS	DeepLabv3+	84.7%	81.2%	68.4%	74.0%	0.99	0.11	0.29
	DeepFill	87.0%	80.6%	78.6%	79.4%	1.09	0.10	0.32
	R & C	**93.2%**	**86.4%**	**93.8%**	**89.8%**	**0.24**	**0.05**	**0.20**
GSV	Pix2Pix	84.3%	73.9%	64.2%	67.3%	0.62	0.08	0.24
	DeepFill	89.7%	75.4%	89.4%	81.6%	0.73	0.07	0.25
	R & C	**94.8%**	**84.1%**	**96.8%**	**89.9%**	**0.22**	**0.06**	**0.15**
GSV	DeepLabv3+	79.7%	60.9%	61.3%	60.8%	0.51	**0.07**	0.28
	DeepFill	82.1%	63.1%	78.9%	70.0%	0.65	0.11	0.35
	R & C	**90.2%**	**75.9%**	**92.2%**	**83.2%**	**0.21**	0.08	**0.22**

Appendix E.3. RFCNet

For partially-occluded facade segmentation with additional views, as shown in Table E.3, our RFCNet achieves better facade results when evaluated for facade correctness and regularization as compared to the segmentation of the first view and the segmentation completed using DeepFill.

Table E.3. ***RFCNet Quantitative Comparison.*** **We compare the initial facade segmentation, the initial segmentation completed by DeepFill, and the output after fusing additional views by applying our whole RFCNet framework in WVS and GSV to the initial segmentation.**

Dataset	Method	Facade Correctness				Facade Regulariz. Error		
		Acc.	Pre.	Rec.	F1	E_a	E_s	E_p
WVS	EncNet	88.4%	89.6%	68.7%	77.6%	0.54	0.08	0.15
	DeepFill	93.7%	89.6%	89.8%	89.5%	0.71	0.09	0.19
	RFC	**96.9%**	**91.4%**	**99.0%**	**95.0%**	**0.23**	**0.05**	**0.10**
WVS	DeepLabv3+	87.0%	82.6%	71.1%	75.8%	0.55	0.09	0.22
	DeepFill	90.7%	85.4%	82.4%	83.7%	0.92	0.15	0.29
	RFC	**95.7%**	**87.2%**	**99.7%**	**93.1%**	**0.23**	**0.08**	**0.19**
GSV	Pix2Pix	85.5%	76.5%	67.3%	70.7%	0.53	0.07	0.19
	DeepFill	90.1%	79.8%	85.4%	82.1%	0.65	0.07	0.17
	RFC	**95.5%**	**86.1%**	**98.0%**	**91.6%**	**0.22**	**0.06**	**0.13**
GSV	DeepLabv3+	81.9%	68.0%	59.2%	62.6%	0.57	**0.08**	0.25
	DeepFill	84.1%	68.1%	76.3%	71.8%	0.75	0.10	0.32
	RFC	**90.1%**	**76.1%**	**91.1%**	**82.9%**	**0.24**	0.09	**0.20**

Appendix F. Additional Urban Structures

We demonstrate the usage of our framework on additional urban structures, including multiple label ECP (Teboul et al., 2011) facades and SpaceNet (Etten et al., 2018) satellite-based building footprints and roofs, as shown in Figure F.1. In the following sections, we describe the details of applying our approach to these patterns.

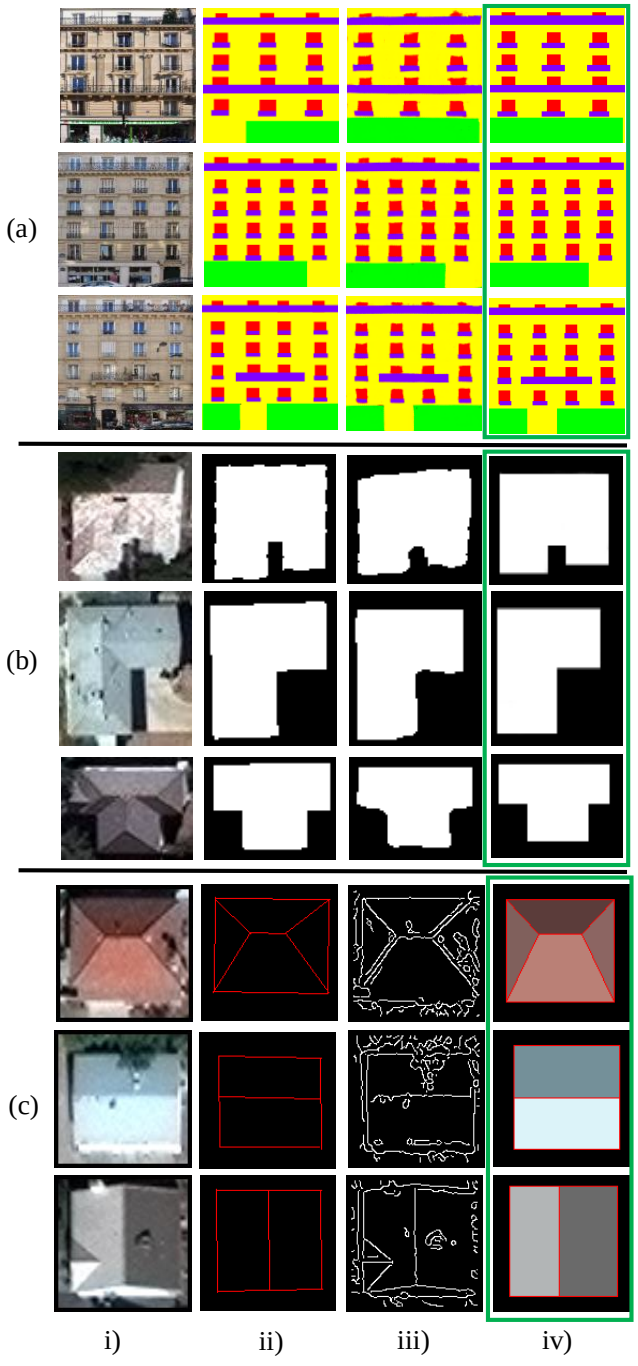

Fig. F.1. ***Additional Urban Structures.*** **We show (a) ECP facades with multiple labels, (b) building footprints, and (c) roofs. For each, i) ECP or SpaceNet provided image, ii) ground truth, iii) initial segmentation or Canny edges, and iv) our RFCNet result. Note: Roof outputs are rendered using the average color.**

Appendix F.1. Multiple Label Facades

Our RFCNet can significantly improve facade correctness and regularization as shown in the comprehensive experiments in the paper. To be able to compare between satellite and street-view facades, we focused on windows. Yet, as seen in Figure F.1 (a), our approach indeed supports multi-label facades.

As in Section 3.1 of our paper, windows usually show regularities as a whole facade (e.g., A_t and A_b for each floor, A_l and A_r for each column of windows). However, based on our observations, balconies (or doors) usually exhibit regularities floor by floor (e.g., A_b and one or multiple groups of A_t for each floor as seen in Figure F.1 (a)).

Following the above assumptions, we add single floor synthetic examples supporting one or multiple groups of A_t (e.g., see Figure F.2 (a)) to the existing synthetic dataset described in Section 3.3 of the paper. After refining our existing models, we generate outputs for windows, balconies and doors separately. Especially for balconies, we get results floor by floor. In the end, we combine all together and get the final output illustrated in Figure F.1 (a). Compared with the initial segmentation, our approach generates regularized and visually pleasant results.

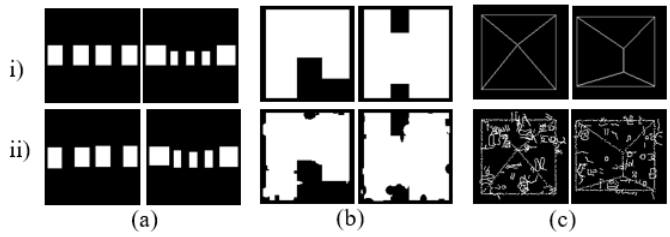

Fig. F.2. ***Data Transformation.*** **We show (a) ECP facades with multiple labels, (b) building footprints, and (c) roofs. For each, i) Clean and regularized synthetic images. ii) Images after corresponding transformation.**

Appendix F.2. Building footprints

Regarding regularities, (man-made) buildings exhibit properties (see Figure F.1 (b)) such as straight walls, parallel walls, walls meeting at one of a set of predetermined angles (e.g., 90 or 135 degrees), symmetrical arrangements, and other features. For the sake of simplicity, we focus on straight walls, parallel walls and right angle regularities. In order to support various building styles (shapes), our synthetic dataset includes the rectangle, L, T, U, Z and H shapes. In addition, in order to represent the noisy and irregular building footprint segmentation, we need to transform our clean/regularized synthetic images (see Figure F.2 (b)). We apply random occlusion or bumps (e.g., different shapes and sizes) around the footprint boundaries. Based on experiments, furthermore, we include different levels of transformations (low-noisy, medium-noisy and high-noisy) when training. Finally, we train our framework and generate the outputs shown in Figure F.1 (b). Our method improves the regularities of footprints (e.g., straight walls, parallel walls and corners with right angles) compared to the initial segmentation.

Appendix F.3. Roofs

As for roof regularities (as seen in a 2D image), two aspects are involved: external edges (e.g., eaves) and internal edges (e.g., ridges and hips). In our experiment, we consider roofs consisting of rectangular components, meaning that the external edges form parts of a rectangle. For internal edges, ridges should follow the main direction of the roof (e.g., parallel or perpendicular to eaves), and hips are connected to ridges. In our current synthetic dataset, we support flat, gable, hip and

pyramid styles. Further, We transform the synthetic roof images by adding random noisy curve lines and randomly remove small parts of the edges (see Figure F.2 (c)). During training, similarly we apply different levels of transformations. For visual performance, we render our results by computing the average color of individual faces, and present example results in Figure F.1 (c).

References

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence .

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

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI .

Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A., 2015. Semantic image segmentation with deep convolutional nets and fully connected crfs. ICLR .

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation, in: ECCV.

Cohen, A., Oswald, M.R., Liu, Y., Pollefeys, M., 2017. Symmetry-aware façade parsing with occlusions, in: 2017 International Conference on 3D Vision (3DV).

Etten, A.V., Lindenbaum, D., Bacastow, T.M., 2018. Spacenet: A remote sensing dataset and challenge series. CoRR abs/1807.01232.

Friedman, S., Stamos, I., 2012. Online detection of repeated structures in point clouds of urban scenes for compression and registration. International Journal of Computer Vision 102, 112–128.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H., 2019. Dual attention network for scene segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn, in: 2017 IEEE International Conference on Computer Vision (ICCV).

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Iglovikov, V., Seferbekov, S., Buslaev, A., Shvets, A., 2018. Ternausnetv2: Fully convolutional network for instance segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K., 2015. Spatial transformer networks. Advances in Neural Information Processing Systems 28 (NIPS 2015) .

Jiang, H., Nan, L., Yan, D., Dong, W., Zhang, X., Wonka, P., 2016. Automatic constraint detection for 2d layout regularization. IEEE Transactions on Visualization and Computer Graphics .

Jie Yang, Zhiquan Qi, Y.S., 2020. Learning to incorporate structure knowledge for image inpainting, in: Proceedings of the AAAI Conference on Artificial Intelligence.

Joo, D., Kim, D., Kim, J., 2018. Generating a fusion image: One's identity and another's shape, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Kelly, T., Guerrero, P., Steed, A., Wonka, P., Mitra, N.J., 2018. Frankengan: Guided detail synthesis for building mass-models using style-synchonized gans. ACM Transactions on Graphics 37.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: Bengio, Y., LeCun, Y. (Eds.), International Conference on Learning Representations, ICLR 2015.

Kozinski, M., Gadde, R., Zagoruyko, S., Obozinski, G., Marlet, R., 2015. A mrf shape prior for facade parsing with occlusions, in: IEEE Computer Vision and Pattern Recognition.

Li, S., Zou, C., Li, Y., Zhao, X., Gao, Y., 2020. Attention-based multi-modal fusion network for semantic scene completion. Proceedings of the AAAI Conference on Artificial Intelligence 34.

Li, Y., Zhang, J., Cheng, Y., Huang, K., Tan, T., 2018. Df2net: Discriminative feature learning and fusion network for rgb-d indoor scene classification, in: Proceedings of the AAAI Conference on Artificial Intelligence.

Liu, H., Zhang, J., Zhu, J., Hoi, S.C.H., 2017. Deepfacade: A deep learning approach to facade parsing, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Mahmud, J., Price, T., Bapat, A., Frahm, J.M., 2020. Boundary-aware 3d building reconstruction from a single overhead image, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Masci, J., Meier, U., Cireşan, D., Schmidhuber, J., 2011. Stacked convolutional auto-encoders for hierarchical feature extraction, in: Proceedings of the 21th International Conference on Artificial Neural Networks (ICANN).

Mathias, M., Martinovic, A., Van Gool, L., 2016. Atlas: A three-layered approach to facade parsing. International journal of computer vision .

Nauata, N., Furukawa, Y., 2020. Vectorizing world buildings: Planar graph reconstruction by primitive detection and relationship inference, in: European Conference on Computer Vision.

Nishida, G., Bousseau, A., Aliaga, D.G., 2018. Procedural modeling of a building from a single image. Computer Graphics Forum (Eurographics) 37.

Olszewski, K., Tulyakov, S., Woodford, O., Li, H., Luo, L., 2019. Transformable bottleneck networks. The IEEE International Conference on Computer Vision (ICCV) .

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in pytorch, in: NIPS-W.

Prabhakar, K.R., Srikar, V.S., Babu, R.V., 2017. Deepfuse: A deep unsupervised approach for exposure fusion with extreme exposure image pairs, in: 2017 IEEE International Conference on Computer Vision (ICCV).

Ronneberger, O., P.Fischer, Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention (MICCAI).

Takikawa, T., Acuna, D., Jampani, V., Fidler, S., 2019. Gated-scnn: Gated shape cnns for semantic segmentation. ICCV .

Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., Paragios, N., 2011. Shape grammar parsing via reinforcement learning, in: 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).

Trinidad, M.C., Martin-Brualla, R., Kainz, F., Kontkanen, J., 2019. Multi-view image fusion, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV).

Tyleček, R., Šára, R., 2013. Spatial pattern templates for recognition of objects with regular structure, in: Proc. GCPR, Saarbrucken, Germany.

Woo, S., Park, J., Lee, J.Y., Kweon, I.S., 2018. Cbam: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision (ECCV).

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S., 2018. Generative image inpainting with contextual attention, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S., 2019. Free-form image inpainting with gated convolution, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

Yu, T., Guo, Z., Jin, X., Wu, S., Chen, Z., Li, W., Zhang, Z., Liu, S., 2020. Region normalization for image inpainting., in: Proceedings of the AAAI Conference on Artificial Intelligence.

Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A., 2018. Context encoding for semantic segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Zhang, X., May, C., Aliaga, D., 2020. Synthesis and completion of facades from satellite imagery, in: Proceedings of the European Conference on Computer Vision (ECCV).

Zheng, C., Cham, T.J., Cai, J., 2019. Pluralistic image completion, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Zheng, Q., Sharf, A., Wan, G., Li, Y., Mitra, N.J., Cohen-Or, D., Chen, B., 2010. Non-local scan consolidation for 3d urban scenes. ACM Transactions on Graphics 29, to appear.

Zhu, M., Pan, P., Chen, W., Yang, Y., 2020. EEMEFN: low-light image enhancement via edge-enhanced multi-exposure fusion network, in: Proceedings of the AAAI Conference on Artificial Intelligence.