Synthesis and Completion of Facades from Satellite Imagery

Xiaowei Zhang(✉), Christopher May, and Daniel Aliaga

Purdue University, West Lafayette, USA
{zhan2597,may5,aliaga}@purdue.edu

Abstract. Automatic satellite-based reconstruction enables large and widespread creation of urban areas. However, satellite imagery is often noisy and incomplete, and is not suitable for reconstructing detailed building facades. We present a machine learning-based inverse procedural modeling method to automatically create synthetic facades from satellite imagery. Our key observation is that building facades exhibit regular, grid-like structures. Hence, we can overcome the low-resolution, noisy, and partial building data obtained from satellite imagery by synthesizing the underlying facade layout. Our method infers regular facade details from satellite-based image-fragments of a building, and applies them to occluded or under-sampled parts of the building, resulting in plausible, crisp facades. Using urban areas from six cities, we compare our approach to several state-of-the-art image completion/in-filling methods and our approach consistently creates better facade images.

Keywords: Image synthesis and completion · Inverse procedural modeling · Satellite imagery

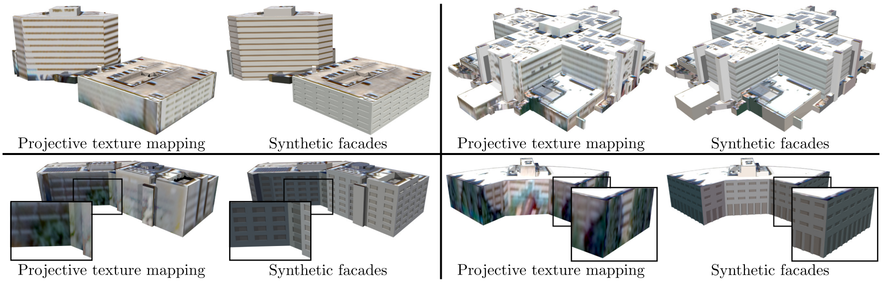
1 Introduction

Urban inverse procedural modeling is beneficial for many simulation, training, and entertainment applications. Using satellite data enables large scale, potentially global reconstructions. However, satellite data is challenging to work with due to limitations in resolution, noise, complex camera models, partial coverage, and occlusions. These aspects hinder high quality urban reconstruction.

Our key observation is that buildings in dense urban areas typically exhibit a regular, grid-like facade structure. We exploit this observation via a machine learning-based inverse procedural modeling approach to determine procedural parameters for a number of facade grammars in the presence of incomplete data. The grammars are then applied to the faces of reconstructed 3D building models during a facade completion phase. This methodology significantly improves the

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58536-5_34) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12347, pp. 573–588, 2020.
https://doi.org/10.1007/978-3-030-58536-5_34



Projective texture mapping Synthetic facades Projective texture mapping Synthetic facades

Projective texture mapping Synthetic facades Projective texture mapping Synthetic facades

Fig. 1. *Examples of facade synthesis and completion.* Our method automatically creates procedural facades from satellite-based images despite noise, occlusions, and incomplete coverage.

resilience to occluded/noisy images and produces more accurate facade layouts as compared to alternative segmentation-based methods. Since satellite images have a very limited off-nadir view (e.g., at most 20 to 40°), and building surface coverage is limited (e.g., the orbital path of the satellite is not able to capture all building sides), often only fragments of a building are seen. Furthermore, facades that are observed may only be seen at very oblique angles, resulting in low resolution and stretched facade images. Nonetheless, a procedural approach has the ability to recreate the observed portion as well as create a plausible synthesized facade reconstruction of the occluded/not-sampled fragments. The result is plausible, complete building facades.

Our approach takes as input 3D building models obtained from point-clouds (e.g., [19]), as well as satellite image fragments projected onto the faces of the building models. The image fragments are used together with trained deep networks to find a representative sample of a facade with minimal noise, and infer its style and procedural parameters. The parameters are then used to complete the rest of the facade, and potentially other non-observed facades of a building. In the end, our approach produces complete facade layouts applied to building models. Figure 1 shows example results of our approach. Since we have a procedural output (instead of an image), we can zoom-in to any part of the facade and still have a crisp result, as observed in the close-up views.

Our results yield improvements over other methods applied to the same data. Over our six test areas, each spanning 1–2 km^2, our method is consistently better than the prior work we compare to quantitatively and qualitatively, and the average accuracy of several performance metrics is 85.4% despite significant occlusions, noise, and strong blurriness. Further, our deep networks are trained on a new dataset of rectified satellite facade views with ground truth segmentation that we also offer as a contribution. As far as we know, our work is the first pipeline to handle façade reconstruction based on satellite imagery despite the occlusions and resolution limitations of such imagery.

Our main contributions include: (1) A machine learning based pipeline addressing occlusion and regularity for satellite facade patterns. (2) A facade completion technique to generate plausible facade layouts based on the predicted grammars and building geometry. (3) A satellite facade dataset with ground truth window and door segmentation.

2 Related Work

Related work can be divided into building-envelope reconstruction, facade reconstruction, and forward/inverse procedural modeling. Musialski et al. [18] provides a review of urban reconstruction. Despite having the highest-resolution commercially available satellite imagery (i.e., WorldView3), the main structure of a building occupies on average 90 × 90 pixels on the ground plane and on average the best observation of a facade is 20 pixels tall. Aside from the relatively low resolution of satellite imagery, there are several other aspects that differentiate satellite-based multi-view stereo reconstruction from ground/aerial multi-view stereo reconstruction [21,22]. First, satellites use scan-line sensors producing images with a different projection model than standard frame cameras. Usually a rational polynomial coefficient (RPC) model is used. Such RPCs are hard to calibrate, require iterative processes, need many ground control points, and performing 3D to 2D as well as 2D to 3D mapping is difficult [34]. Second, the image quality can vary a lot due to a number of factors, including the viewing angles of satellite sensors are greatly limited by the orbit (i.e., not very off-nadir), images of an area might be days/weeks/months apart yielding different illumination and potentially physical changes, and radiometric quality is lower despite attempts of atmospheric corrections (see Fig. 2). While our work does not address the problem of 3D building reconstruction, building geometry is reconstructed automatically from a SOTA multi-view stereo point cloud obtained from satellite images, similar to and by extending [13,32]. It's important to note that the above limitations affect the quality of the reconstructed models, which are used by our facade synthesis method. Thus we cannot expect to have perfect building geometry with which to produce synthetic facade layouts.

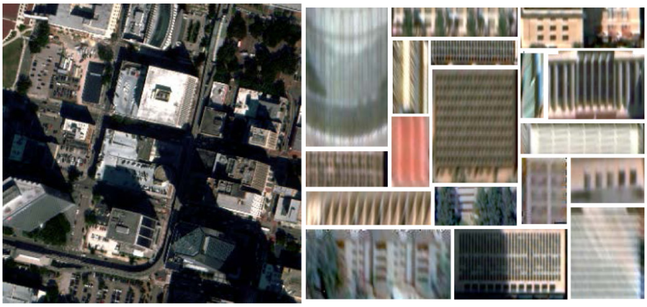


Fig. 2. *Satellite image and facade closeups.* Example satellite image and views of some typical facades.

Almost all facade reconstruction methods use ground or aerial imagery, typically rectified and rectangular. Many approaches have been followed (e.g., using dynamic programming [3], using lattices [23], using matrix approximations [29], and inferring grammars from pre-labelled segments [7,12,15]). However, these methods do not perform well for our very under-sampled facades. For example, see our comparisons in the results section.

More recently, deep learning based facade parsing has obtained excellent results for ground-level imagery. For example, Liu et al. [14] and Fathalla et al. [6] perform facade segmentation but assume high-resolution frontal views. Nishida et al. [20] further assumes hand-specified building silhouettes and their facade stage depends on having clear boundaries between floors and between columns. Further, none of these account for the significant occlusions in satellite-based facades. Kelly et al. [10] could automatically and realistically decorate buildings by synthesizing geometric details/textures. However, their work requires style references (e.g., façade and roof textures, window layouts) and such references from satellite would be very low-resolution and heavily occluded. Kozinski et al. [11] (and partially Mathias et al. [16]) include provisions for occlusions but depend on many assumed structural priors for numerous object classes and SIFT feature vectors. On average the facades we encounter are only 20 × 90 pixels in size (often significantly worse) and thus make it prohibitive to determine such detailed structure. Image-to-image translation, such as Isola et al. [9] and Zhu et al. [35], has been proposed but does not support all of regularity, occlusions, and satellite data. From the semantic segmentation point of view, facade parsing could also be considered as a segmentation task. Many papers (e.g. DeepLabv3+ [2], EncNet [31], etc.) have shown great success with segmentation, but none of them use satellite facade data. Thus we trained those neural networks from scratch using our created satellite facade dataset (see Results section) and observe that these state-of-the-art segmentation neural networks also suffer from the low-quality of satellite facade data and cannot generate crisp facades.

Filling-in missing pixels of an image, often referred as image in-painting or completion, is an important task in computer vision. Deep learning and GAN-based approaches (e.g., DeepFill [30], PICNet [33]) have achieved promising results in this task. However, image in-painting is ill-suited for resolving shadows and occlusions in satellite facade images. First, detection of these areas is a very challenging problem, especially for satellite data. Second, even assuming these areas could be detected automatically, image in-painting approaches cannot infer correctly due to the low quality of satellite facade data. We also show in the Results section comparisons to these approaches.

Inverse procedural modeling (IPM) attempts to determine the procedure (e.g., rules and/or parameter values) yielding a desired geometric output. IPM has been used to stochastically derive a procedural model [24,26], infer Manhattan-world buildings from aerial imagery [28], or arbitrary buildings from polygonal data [1,4,5]. However, none of these methods have been used to infer building facade layouts from satellite data.

3 Facade Synthesis

While there might be 1–20 satellite images observing portions of buildings, there is usually not a high quality satellite observation of every facade on a building due to shadows, foliage/occlusions, and limited resolution. Thus simply applying satellite images to building faces via projective texture mapping is inadequate. Further, such texture mapping depends on very accurate image-to-image registration, geometric modeling, and complete coverage of all building facades. Our approach attempts to overcome these issues by synthesizing procedural facades using a selected subset of the available satellite imagery, and then applying these facades across the entire building. This approach has the following advantages:

- *Crisp Results.* The produced facade details will be crisp and visible at any resolution.
- *Exploits Best Observations.* Without relying on accurate RPCs and image registration, we choose the best, potentially fragmented, observations of each building and use it to obtain facade details.
- *Completes Missing Fragments.* Even if a facade/fragment is missing, we can fill-in the facade with details from a partial observation (or in worst case with details from neighboring facades).

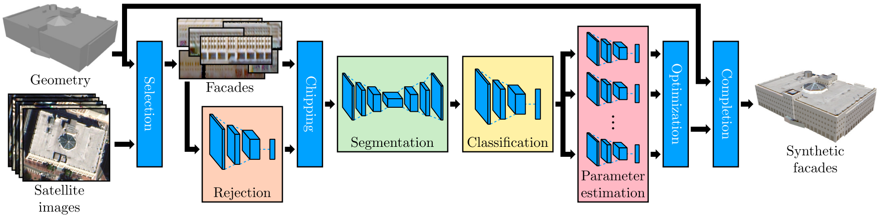


Fig. 3. *Pipeline.* The pipeline of our multi-stage approach for facade completion and synthesis.

We provide an overview of the proposed procedural facade approach in Fig. 3 and in the following we describe the pipeline starting with our selection method, followed by our deep-learning based facade style classification and parameter estimation, and finally our facade and building completion.

3.1 Selection

In a first stage, we choose the satellite image that has low grazing angle and does not have much dark pixels as the best view of the facade, and the resulting image is used as input to the rest of the pipeline. In many cases, even the best observation of a facade is not useful due to noise, shadows, trees, and occlusions.

Thus we employ a deep-learning based rejection model to prevent further processing of any such facades. Rejected facades will not undergo classification or parameter estimation, but can still receive synthetic facade layouts as part of the completion phase (Sect. 3.3).

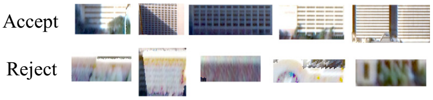


Fig. 4. *Accept or reject.* The first row shows facades that our rejection model will accept. The second row shows facades that will be rejected.

Our rejection network is based on a pre-trained ResNet [8] model, in which we modify the last fully connected layer to two classes: one for "good" facades to be accepted and the other for "bad" facades to be rejected. We used 120 examples of "good" facades from our facade data set and 120 examples of "bad" facades, resulting in 1920 training images in total after applying data augmentation such as flip, rotation, random crop and intensity variations. The model performs with 92% accuracy when tested on 200 test images. Figure 4 shows some examples of accepted and rejected facades.

3.2 Classification and Parameter Estimation

In a second stage, our approach estimates the style and parameters of an equivalent procedural facade representation. Our method extracts a "chip" from the selected facade image because i) satellite-based images often suffer from occlusions and thus assuming a full facade view would be prohibitive, and ii) otherwise the parameter space would be unnecessarily large as the number of floors/windows may vary significantly yet the spacing between floors and windows is regular. The procedural representation for the entire facade is obtained from the chip and then used during the next stage to complete each facade.

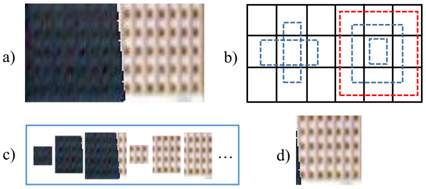


Fig. 5. *Chip extraction.* a) Original facade. b) Division of a) into tiles and demonstration of how chips are formed. c) Apply b) to a). d) The best chip.

Chip Extraction. To choose the best chip to extract, we divide the original facade image into a set of N tiles each of size 6×6 m. Each chip is formed by selecting a tile as the center and then varying the chip size to 6, 12, or 18 m and varying the aspect ratio (e.g., 1:1, 1:2, or 2:1). In total, $9N$ different candidate chips are produced for each facade. Please see Fig. 5 for a visual depiction. We evaluate each chip by passing it through our rejection network and evaluating its rejection score. The chip with the lowest rejection score is considered to be the cleanest chip found for the facade, and is selected to represent this facade further in the pipeline.

Segmentation. During segmentation, we only label each pixel as belonging to window/door or non-window/non-door since other facade classes are usually not visible in satellite imagery. During development, we experimented with several state-of-the-art deep-network based semantic segmentation models (e.g., DeepLabv3+ [2], EncNet [31], and Pix2Pix [9]). Please see Segmentation models in the Results section for quantitative and qualitative comparisons among these architectures. We found that the architecture of Pix2Pix [9] performs among the best ones, and in particular we specify the generator architecture to consist of ResNet blocks, the discriminator architecture to be 34×34 PatchGAN, and the input image size to be 96×96. We train the segmentation network from scratch using our own manually created satellite facade dataset. Specifically, we train with 120 facade images (960 after applying the aforementioned data augmentation) along with ground truth from our dataset.

After segmentation, we have binary segmented chip facades with two labels: one representing windows and doors (black), and one representing the building wall (white). Using a binary representation eases the burden for deep-network based recognition and parameter estimation. In addition, we apply some image processing techniques to further refine the segmented image. First we perform a small amount of dilation (e.g. rectangular dilation with a kernel size of 3 pixels) to reduce some of the noisy black window/door pixels. Next, since some facades are not perfectly rectified (due to errors in image registration and/or geometry), we perform a global image rotation computed automatically to force rows of windows/doors to be horizontal. Further, each window/door is replaced by a filled-in version of its rectangular bounding box. The end result is a binary image with rectangular windows and doors representing the facade, and serves as the input to our recognition and estimation networks.

Grammar Classification and Estimation. We represent a synthetic facade by one of six possible grammars each with a number of parameters, defined in a systematic fashion. While a single grammar with many parameters might be able to express more facades we found its generality to result in overall lower quality given the low-resolution nature of our facade imagery. For our grammar classification, a facade may contain doors and windows, or only windows. Further, the windows can be arranged as a grid of disjoint windows, as columns of vertically abutting windows, or as rows of horizontally abutting windows (see Fig. 2 and Fig. 6). Since window shapes are hard to differentiate with satellite data, we treat all windows as rectangles.

Which grammar a facade belongs to, along with the parameters for said grammar, is determined with a set of deep networks based on ResNet [8]. There is a classification network, which determines the grammar, followed by six parameter estimation networks, for determining the parameters specific to each grammar. The classification network is a ResNet [8] with modification of the last fully-connected layer to the number of grammars. The final output layer of this network yields confidence values for each of the aforementioned grammars. After classifying a facade via this network, the segmented facade chip is then sent through the parameter estimation network that corresponds to the highest confidence value in the classification output.

To robustly find the procedural parameters for the classified grammar, we use a separate deep network for each individual grammar, all of which are also based on ResNet [8]. They differ only in the last fully-connected layer, where we modify the number of parameters to match that of the grammar. We also use mean squared error as the loss function for our estimation networks. The predicted parameters (e.g., window rows, columns, relative size, etc.) altogether yield a synthetic facade that is similar to the input image.

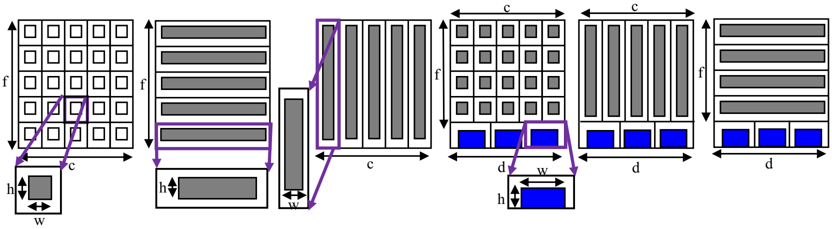


Fig. 6. *Grammars.* Our grammars of (1–3) three styles of only windows and (4–6) three styles with doors at the base. “f” stands for the number of floors. “c” is the number of column boundaries. “d” is the number of doors. “h” is the relative height and “w” is the relative width. Please see the close-ups for additional parameters in the different grammars.

To train the estimation networks by systematically iterating over possible facade parameter configurations, we synthesized 200,000, 20,000, 20,000, 400,000, 50,000, and 50,000 facades from grammars 1 to 6 in Fig. 6, respectively, based on the different number of parameters for each. We also perform data augmentation accounting for noise and errors in the segmentation (i.e., up to 10% noise such as perturbation of boundaries in windows/doors) and randomly remove up to 10% of windows/doors. To train the classification network, we collected 108,000 images in total from the aforementioned training images, distributed evenly among all six grammars.

Optimization. After recognition and parameter estimation, we perform a coarse-to-fine refinement for each chip. Segmentation suffers from noise, shadows, trees, and occlusions. Fortunately, our parameter estimation network is able to recover a procedural facade that fills-in occluded content though there might be an overall translation or scale error. Thus, we define an objective function, using F-score [25], as:

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall} \qquad P^* = \underset{P}{\mathrm{argmax}} F, \qquad (1)$$

In the above, P stands for the grammar parameters in Fig. 6, P^* is the optimal parameter set, *Accuracy* is the percentage of pixels labelled accurately, *Precision* and *Recall* are computed by considering the label windows/doors as positive and the label wall as negative. *Precision* is the number of true positives divided by the sum of true and false positives (e.g., how correct is the windows and doors labelling in our results). *Recall* is the number of true positives divided by the sum of true positives and false negatives where for false negatives we use the number of incorrectly labeled wall pixels (e.g., how many windows and doors pixels our result can correctly label). Overall, F is essentially the harmonic mean of *Precision* and *Recall*.

Our optimizer tries to maximize this function using Monte Carlo stochastic optimization (e.g. altering P such as the number of floors, windows and window size) so as to create a synthetic facade that improves the F-score with respect to the segmentation result. Please see Optimization in Results section for details and comparisons.

3.3 Completion

In a third and final stage, our method applies the estimated procedural parameters to all facades and generates windows and doors with the estimated sizes and spacing. Although the prior step determined parameters for rectangular chips, the actual facades on the buildings are not limited to rectangles but instead may have irregular shapes. To this end, we logically divide a building facade into a set of horizontally-adjacent rectangular sections. Since doors only appear at the bottom of a facade, we partition each rectangular section, that touches ground level, into two subsections: a door subsection extending from the bottom of the facade up to the door height, and a window subsection covering the remainder. Doors are placed horizontally-centered in the door subsections and sized according to the estimated parameters. The window subsections are then further subdivided into window cells, also sized and spaced according to the estimated parameters, with one window placed into each cell. The tallest window subsections determine vertical window placement such that building floors are level across all sections.

Since each chip's parameters are estimated independently, neighboring facades will in general have different door/window sizes and spacing, and potentially different grammars. To remedy this issue, we first group facades together based on similar heights. All facades within each group are then forced to use the grammar of the highest scoring facade in the group, scored according to the grammar classification confidence value from the previous stage, with parameter values averaged over matching grammars in the group.

The resulting facades have windows and doors, which are colored according to the average window/door color as determined by the segmentation. Similarly, the facade wall is colored according to the average non-window color.

4 Results

Our method is implemented using OpenCV, OpenGL, and PyTorch, and it runs on an Intel i7 workstation with NVIDIA GTX 1080 cards. We have applied our method to six test areas in the United States captured by WorldView3 satellite images: a portion of (A1) Jacksonville, Florida (2.0 km^2), (A2) UC San Diego, California (1 km^2), (A3) San Fernando, California (1 km^2), (A4) Omaha, Nebraska (2.2 km^2), (A5) San Diego, California (1.2 km^2) and (A6) USC, California (2 km^2). Collectively, the areas have a few hundred buildings and medium to tall buildings and have from 20 to a few hundred windows/doors each. Our method runs automatically yielding facades for 14 buildings per minute. The training time for our classification network is about 12 h, and the training time for our estimation networks from grammars 1) to 6) is about 20 h, 3 h, 3 h, 36 h, 8 h, and 8 h, respectively.

Dataset. In order to train our neural network models, evaluate our method, and compare with other methods, we present a dataset of real satellite facades, which includes about 400 rectified images of facades from the aforementioned six areas, which have been manually annotated with two different labels: one for windows/doors and the other for the walls. Because of the low-quality of these facades, even humans can't precisely do the segmentation. Thus, missegmentation and misalignment always exist. Further, we carefully refine the annotations for 61 facade images and use those facades as a test data set for evaluating models/methods.

Pipeline Steps. We show example pipeline steps in Fig. 7 which includes chip extraction results, segmentation results, image processing results and our final facade completion results. Additional example facades are in supplemental figures. Our paper video also shows the pipeline and example results.

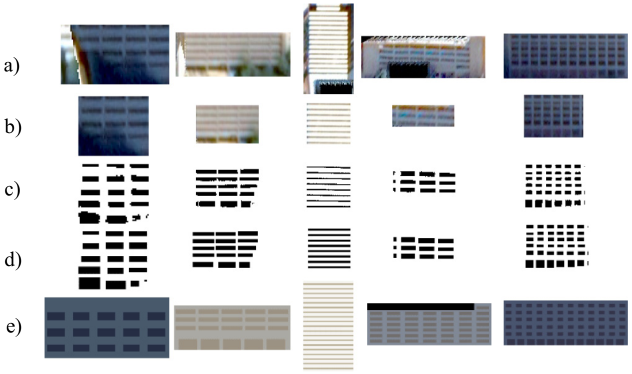


Fig. 7. *Pipeline steps.* a) Selected facade images. b) Facade chips. c) Results of using our segmentation model b). d) Images after applying dilation, rotation and replacement of windows/doors with filled-in rectangular bounding boxes and then being fed to our neural networks. e) Synthesized facades.

Table 1. *Segmentation quantitative comparison.* Pixel accuracy, precision, recall and F-score metrics evaluated on 61 facades for models from b) to g). Those terms are defined in optimization section.

Model	Accuracy	Precision	Recall	F-score
b)	0.843665	0.756	0.747	0.742
c)	0.8482	0.795	0.712	0.742
d)	0.866343	0.836	0.741	0.771
e)	0.846425	0.802	0.696	0.732
f)	0.849911	0.776	0.725	0.740
g)	0.870966	0.864	0.709	0.766

Segmentation Models. We test satellite facade segmentation on three state-of-the-art neural network architectures: Pix2Pix [9], Deep Labv3+ [2] and EncNet [31]. We train these architectures from scratch using our data set and also customize the hyper-parameters to fit our segmentation problem. For Pix2Pix we also try different generator and discriminator architectures which could support different sizes of input images. See supplemental table and supplement Fig. 2 for specific configurations and qualitative comparisons. Please see Table 1 for quantitative comparisons. Based on this comparison, we perceive Pix2Pix_96 to work best and it is the segmentation model we use in our approach.

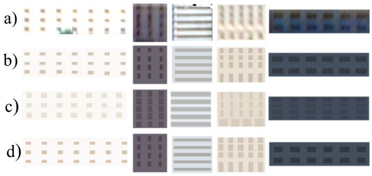


Fig. 8. *Optimization qualitative results.* a) Original facades. b) Manually created ground truth. c) Our results without optimization. d) Our results with optimization.

Table 2. *Optimization quantitative comparison.* Pixel accuracy, precision, recall, F-score and blob accuracy evaluated on 61 facades for models c) and d) in Fig. 8.

Method	Accuracy	Precision	Recall	F-score	Blob
c)	0.725	0.556	0.673	0.597	0.810
d)	0.880	0.818	0.834	0.815	0.923

Optimization. We evaluate 61 facade images using both our method without optimization and our method with optimization. Thus we show that we improve pixel accuracy, precision, recall, F-score and blob accuracy by perturbing grammar parameters. The blob accuracy is the window count accuracy defined as:

$$Blob = 1 - \frac{|Our\_Window\_Count - Ground\_Truth\_Window\_Count|}{Ground\_Truth\_Window\_Count}, \tag{2}$$

Please see Fig. 8 and Table 2 for qualitative and quantitative comparisons. In summary, with optimization our metrics improve from 0.69 to 0.85, an improvement of 16% on average.

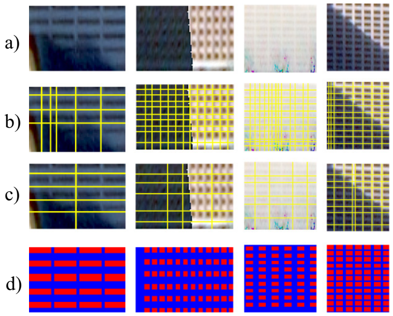


Fig. 9. *Facade subdivision comparison.* We provide a) satellite-based facades to b) an image-based approach, c) Nishida et al. [20], and d) Ours.

Table 3. *Facade quantitative comparison.* We evaluate Mean Absolute Error (MAE) and Mean Relative Error (MRE) of the number of floors and the number of windows per floor on 61 facades for c) and d) in Fig. 9.

Method	MAE		MRE	
	#floors	#windows	#floors	#windows
c)	0.770	0.770	15.8%	12.1%
d)	0.246	0.164	4.2%	3.9%

Comparisons. We compare our approach to several state-of-the-art methods. First, in Fig. 9 we show a visual comparison between the facade subdivision of b) an image-gradient-based approach (e.g., [17]), c) Nishida et al. [20] (retrained using the same training set as our approach), and d) our method. We highlight that Nishida et al. [20] (and also Teboul et al. [27]) essentially make use during their processing pipeline of an image-gradient based method similar to [17] (thus we include the image-gradient comparison). We also include facade quantitative comparisons in Table 3.

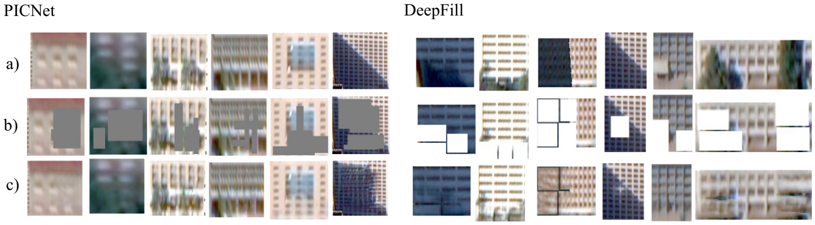


Fig. 10. *Image in-painting.* a) Original facades. b) Rectangular areas to be filled-in. c) Results after inpainting.

Second, we test two state-of-the-art neural network architectures for image inpainting/completion: DeepFill [30] and PICNet [33]. With DeepFill determining which part to “fill” is an unaddressed challenge and thus for this comparison we manually select occluded, shadowed and/or tree-covered areas. In PICNet, we use the random rectangular mask generation method they provide (e.g., select a sufficient number of rectangles within the image to most likely performed all necessary in-filling). Please see Fig. 10 for visual results. While the methods are able to place content in the occluded areas, there are still significant artifacts which will hinder subsequent facade process.

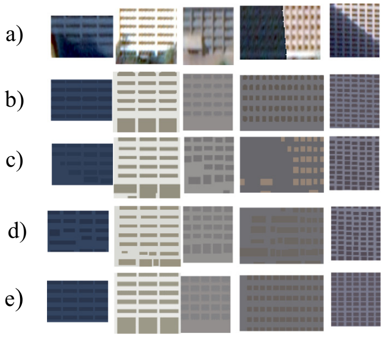


Fig. 11. *Facade comparisons.* Comparison to SOTA methods on facade parsing. a) Input satellite facades. b) Manually created ground truth. c) The results of applying Pix2Pix_96 to a). d) The results of applying Pix2Pix_96 to image completed by DeepFill [30]. e) Ours.

To evaluate the facade processing ability directly using the segmentation model and image in-painting model, we evaluate performance using our 61 test images qualitatively and quantitatively. To be specific, for the segmentation model, we choose the aforementioned Pix2Pix_96 and apply it to the facade images directly. Then, we dilate each window/door to occupy a rectangular bounding box. For the image in-painting model, we choose DeepFill [30] and complete the facade images with manually selected masks. Then we apply the segmentation model to the completed facade images and we also use a version of the windows/doors dilated to rectangles. The quantitative metrics include pixel accuracy, precision, recall, and blob accuracy. In Fig. 11 and Table 4, we show details of comparing our method to the segmentation model and the image in-painting model.

Table 4. *Quantitative comparison.* Pixel accuracy, precision, recall, F-score and blob accuracy evaluated for models from c) to e) in Fig. 11. We evaluated c) and e) on 61 facades in the left table. However the right table shows applying d) to 22 facades (22 out of 61 facades are occluded and suitable for image in-painting.) and we manually set the mask as best as possible.

Method	Accuracy	Precision	Recall	F-score	Blob
c)	0.835	0.695	0.868	0.758	0.891
e)	0.880	0.818	0.834	0.815	0.923

Method	Accuracy	Precision	Recall	F-score	Blob
c)	0.802	0.705	0.797	0.728	0.840
d)	0.806	0.803	0.612	0.677	0.875
e)	0.843	0.768	0.828	0.783	0.918

Examples. Finally, we show in Fig. 12 many close-ups of reconstructed buildings as well as an overall view of one area (A1). Views of our additional areas (A2) and more buildings are in supplemental figures.

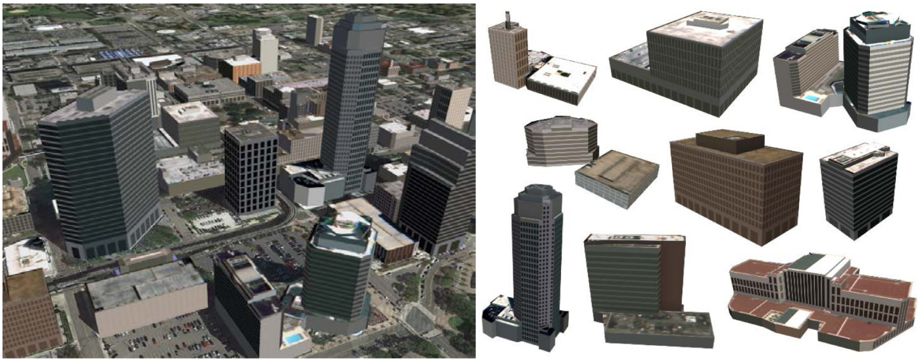


Fig. 12. *Examples.* We show a view of a reconstructed area A1 within Google Earth and close-ups of our buildings.

5 Conclusions and Future Work

We have presented a method to automatically synthesize crisp and regular building facades from satellite imagery. Facades are classified into one of several procedural grammars, and the corresponding parameters are estimated using trained neural networks. The resulting grammars are applied to building models, resulting in complete, plausible facades that are free of the noise, occlusions, and partial coverage that is inherent in satellite data. Our comparisons to other approaches shows the improvement of our method. However, our approach has some limitations. First, for facades whose styles are outside our defined grammars, we could give our best guess. Second, for facades with logos, we didn't show those areas.

Our approach has several avenues of future work. First, we would like to incorporate more general grammar sets to capture finer details. Second, we would also like to incorporate a more sophisticated wall/window color treatment. Finally, we are also interested in estimated and procedural facade textures to give the resulting buildings more details.

Acknowledgements. This research was supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00280. Additional support came from National Science Foundation grants #10001387 and #1835739.

References

1. Bokeloh, M., Wand, M., Seidel, H.P.: A connection between partial symmetry and inverse procedural modeling. ACM Trans. Graph. **29** (2010)
2. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 833–851. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_49

3. Cohen, A., Schwing, A.G., Pollefeys, M.: Efficient structured parsing of facades using dynamic programming. In: IEEE Computer Vision and Pattern Recognition, pp. 3206–3213 (2014)
4. Demir, I., Aliaga, D.G., Benes, B.: Procedural editing of 3D building point clouds. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 2147–2155, December 2015. https://doi.org/10.1109/ICCV.2015.248
5. Demir, I., Aliaga, D.G., Benes, B.: Coupled segmentation and similarity detection for architectural models. ACM Trans. Graph. **34**(4), 1–11 (2015)
6. Fathalla, R., Vogiatzis, G.: A deep learning pipeline for semantic facade segmentation. In: Proceedings of the British Machine Vision Conference 2016, BMVC 2017, September 2017. c 2017. The copyright of this document resides with its authors. It may be distributed unchanged freely in print or electronic forms. http://publications.aston.ac.uk/id/eprint/31805/
7. Gadde, R., Marlet, R., Paragios, N.: Learning grammars for architecture-specific facade parsing. Int. J. Comput. Vis. **117**(3), 290–316 (2016). https://doi.org/10.1007/s11263-016-0887-4
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015). http://arxiv.org/abs/1512.03385
9. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: IEEE Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
10. Kelly, T., Guerrero, P., Steed, A., Wonka, P., Mitra, N.J.: FrankenGAN: guided detail synthesis for building mass-models using style-synchonized GANs. ACM Trans. Graph. **37**(6) (2018). https://doi.org/10.1145/3272127.3275065
11. Kozinski, M., Gadde, R., Zagoruyko, S., Obozinski, G., Marlet, R.: A MRF shape prior for facade parsing with occlusions. In: IEEE Computer Vision and Pattern Recognition, pp. 2820–2828 (2015)
12. Koziński, M., Obozinski, G., Marlet, R.: Beyond procedural facade parsing: bidirectional alignment via linear programming. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9006, pp. 79–94. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16817-3_6
13. Leotta, M.J., et al.: Urban semantic 3D reconstruction from multiview satellite imagery. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2019
14. Liu, H., Zhang, J., Zhu, J., Hoi, S.C.H.: DeepFacade: a deep learning approach to facade parsing. In: International Joint Conference on Artificial Intelligence, pp. 2301–2307 (2017)
15. Martinovic, A., Van Gool, L.: Bayesian grammar learning for inverse procedural modeling. In: IEEE Computer Vision and Pattern Recognition, pp. 201–208 (2013)
16. Mathias, M., Martinović, A., Van Gool, L.: ATLAS: a three-layered approach to facade parsing. Int. J. Comput. Vis. **118**(1), 22–48 (2016). https://doi.org/10.1007/s11263-015-0868-z
17. Müller, P., Zeng, G., Wonka, P., Van Gool, L.: Image-based procedural modeling of facades. ACM Trans. Graph. **26**(3), 85–es (2007). https://doi.org/10.1145/1276377.1276484
18. Musialski, P., Wonka, P., Aliaga, D.G., Wimmer, M., Van Gool, L., Purgathofer, W.: A survey of urban reconstruction. Comput. Graph. Forum **32**, 146–177 (2013)
19. Nguatem, W., Mayer, H.: Modeling urban scenes from pointclouds. In: IEEE International Conference on Computer Vision, pp. 3837–3846 (2017)
20. Nishida, G., Bousseau, A., Aliaga, D.G.: Procedural modeling of a building from a single image. Comput. Graph. Forum **37**, 415–429 (2018)

21. Ozcanli, O.C., Dong, Y., Mundy, J.L., Webb, H., Hammoud, R., Tom, V.: A comparison of stereo and multiview 3-D reconstruction using cross-sensor satellite imagery. In: IEEE Computer Vision and Pattern Recognition Workshops, pp. 17–25 (2015)
22. Qin, R.: Automated 3D recovery from very high resolution multi-view satellite images. In: ASPRS (IGTF) Annual Conference, p. 10 (2017)
23. Riemenschneider, H., et al.: Irregular lattices for complex shape grammar facade parsing. In: IEEE Computer Vision and Pattern Recognition, pp. 1640–1647 (2012)
24. Ritchie, D., Mildenhall, B., Goodman, N.D., Hanrahan, P.: Controlling procedural modeling programs with stochastically-ordered sequential Monte Carlo. ACM Trans. Graph. **34**(4), 1–11 (2015)
25. Sasaki, Y.: The truth of the f-measure. Teach Tutor Mater, January 2007
26. Talton, J.O., Lou, Y., Lesser, S., Duke, J., Měch, R., Koltun, V.: Metropolis procedural modeling. ACM Trans. Graph. **30**(2), 1–14 (2011)
27. Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., Paragios, N.: Shape grammar parsing via reinforcement learning. In: IEEE Computer Vision and Pattern Recognition, pp. 2273–2280 (2011)
28. Vanegas, C.A., Aliaga, D.G., Beneš, B.: Building reconstruction using manhattan-world grammars. In: IEEE Computer Vision and Pattern Recognition (2010)
29. Yang, C., Han, T., Quan, L., Tai, C.L.: Parsing façade with rank-one approximation. In: IEEE Computer Vision and Pattern Recognition, pp. 1720–1727 (2012)
30. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. CoRR abs/1801.07892 (2018). http://arxiv.org/abs/1801.07892
31. Zhang, H., et al.: Context encoding for semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
32. Zhang, X., May, C., Nishida, G., Aliaga, D.: Progressive regularization of satellite-based 3D buildings for interactive rendering. In: Symposium on Interactive 3D Graphics and Games, I3D 2020. Association for Computing Machinery, New York (2020)
33. Zheng, C., Cham, T.J., Cai, J.: Pluralistic image completion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1438–1447 (2019)
34. Zheng, E., Wang, K., Dunn, E., Frahm, J.M.: Minimal solvers for 3D geometry from satellite imagery. In: IEEE International Conference on Computer Vision, pp. 738–746 (2015)
35. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision, pp. 2223–2232 (2017)