

Procedural Modeling of a Building from a Single Image

Gen Nishida¹, Adrien Bousseau², Daniel G. Aliaga¹

¹Purdue University ²Inria

Supplemental Material

1 Building Mass Grammars

Figure A shows the parameters of our first 6 building mass grammars. We define the parameters of other building mass grammars in a similar manner.

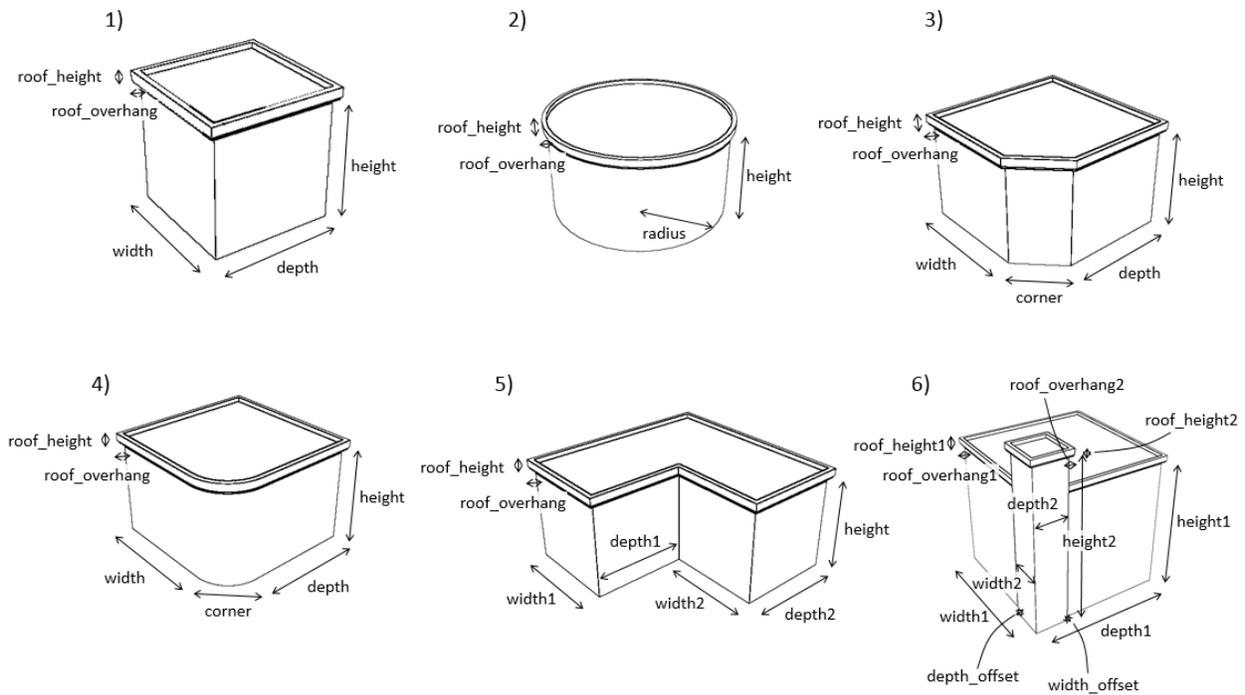


Figure A: The parameters of our first 6 building mass grammars.

2 Façade Grammars

Figure B shows the parameters of our first 4 façade grammars. The same color of windows indicate the same non-terminal. We define the parameters of other façade grammars in a similar manner.

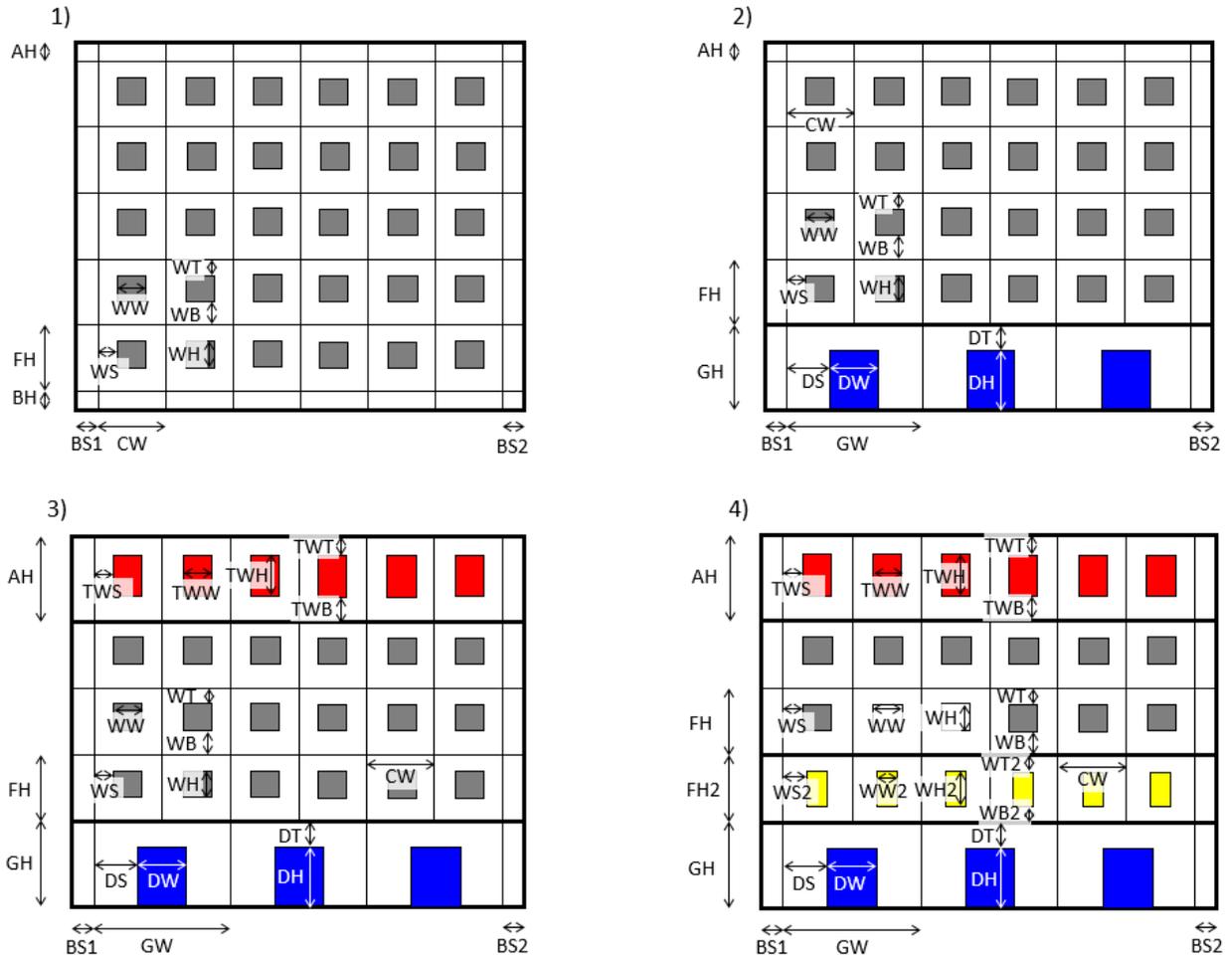


Figure B: The parameters of our first 4 façade grammars.

3 CNN Architecture

For the recognition CNNs, we use the AlexNet architecture (Figures C). For the parameter estimation CNNs, we modify the AlexNet architecture by removing the softmax layer (Figure D).

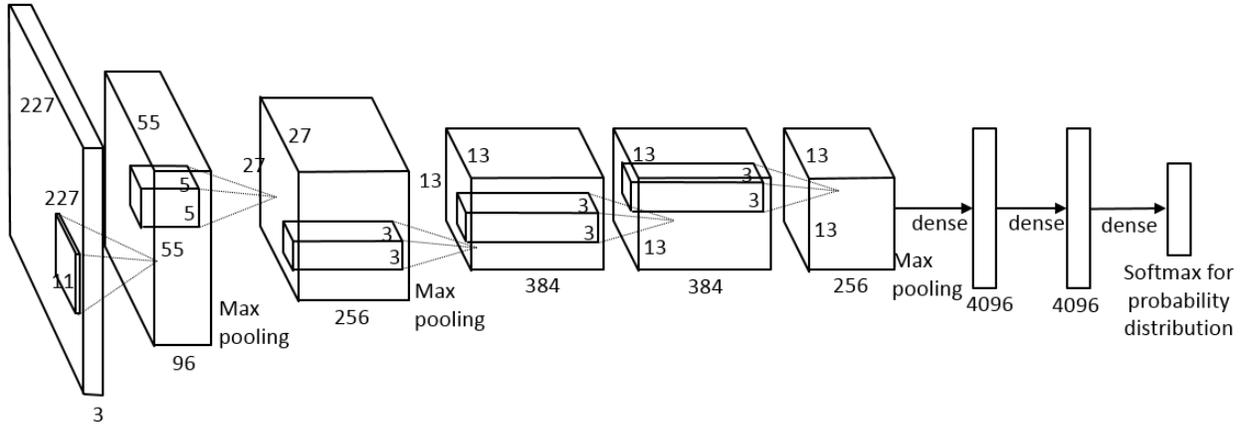


Figure C: The CNN architecture for the recognition CNN.

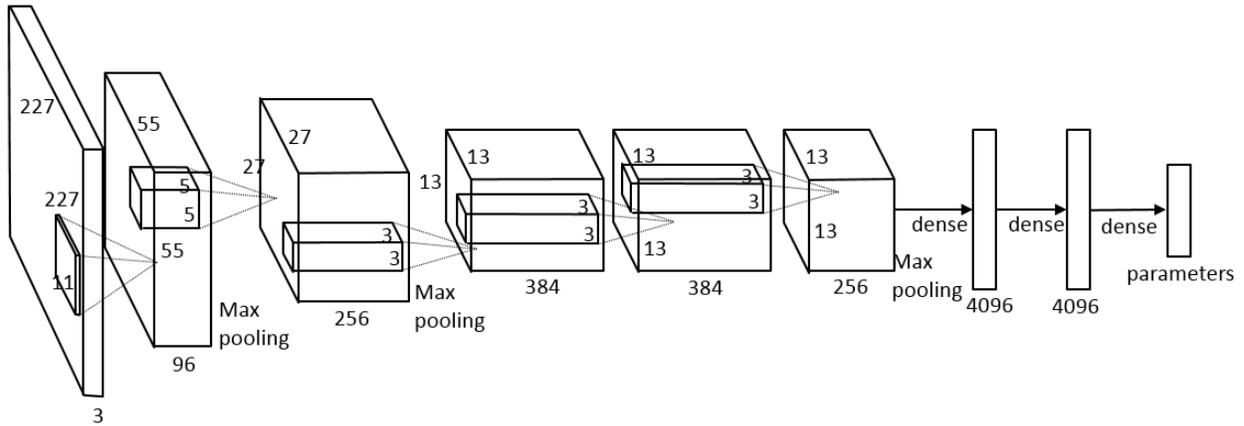


Figure D: The CNN architecture for the parameter estimation CNN.

4 Façade Rectification

Since our grammars for building mass include curved shapes (e.g., cylinders), we in general assume a façade can be a curved face. Thus, we define $Facade = \{f_i\}$ as a set of one or more planar rectangles f_i in 3D space and v_{ij} ($j = 1, \dots, 4$) is the j -th vertex of f_i . Without a loss of generality, we assume that v_{i1} is the bottom left corner of f_i and the vertices are ordered in a counter-clockwise order. Then, for each face f_i , the dot product of the normal vector of the face and the vector to the camera is computed to evaluate if the face is visible. If the angle between the vectors is less than 60 degrees, the face is considered visible and the portion of the image inside the projected face polygon is rectified. Let $p_{ij} = Proj(v_{ij})$ be the projected coordinates of v_{ij} , which can be computed by using the estimated projection matrix \hat{M} . The width and height of rectangle f_i can be computed as:

$$\begin{cases} width = \|v_{i2} - v_{i1}\| \\ height = \|v_{i3} - v_{i2}\| \end{cases}$$

By defining the local coordinate system for f_i such that its origin is located at v_{i1} , its x axis lies along the bottom edge of f_i , and y axis lies along the left side edge, the 2D coordinates of the four vertices in the local coordinate system, p'_{ij} , can be computed. Then, altogether there are four correspondences from p_{ij} to p'_{ij} , so we can compute a per-facade transformation matrix T such that $[p'_{ij}w, w]^T = T[p_{ij}, 1]^T$, where w is a scale factor. The image region inside the projected face polygon is rectified by T . Finally, by combining all the rectified visible faces, the final planar or curved rectified façade image is generated. Note that all the rectified façades have the same height $\|v_{i3} - v_{i2}\|$, so merging the rectified faces is straightforward.

5 Floor and Column Boundaries

[Muller et al. 2007] performs an exhaustive search to find the actual floor and column boundaries among the many local minima of $V(y)$ and $H(x)$. Figure E visualizes $V(y)$ and $H(x)$ on the right and bottom of each façade image, while the red lines represent the local minima. Among many local minima, we want to select the yellow lines as the actual floor and column boundaries, but the task is very challenging without a good approximation of the floor height and column width.

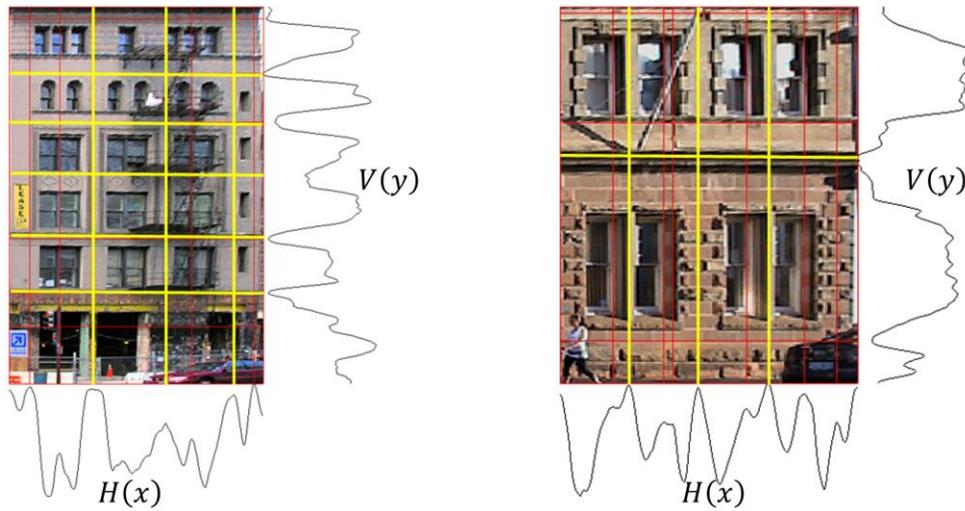


Figure E: Visualization of $V(y)$ and $H(x)$. $V(y)$ and $H(x)$ are visualized on right and bottom, respectively. Local minima of the curves are the candidates for horizontal and vertical boundaries, and yellow lines represent the selected ones.

6 Complete Set of Results for Figure 9

Figures F, G, and H show the first 50 office buildings of ImageNet and the first 30 office buildings of SUN excluding aerial images and images that show only a part of the building. We also show the 3D models generated by our approach. While our approach does not support some complex building shapes (e.g., Figures F-1, F-23, G-32), most of the prominent characteristics of the buildings are captured.

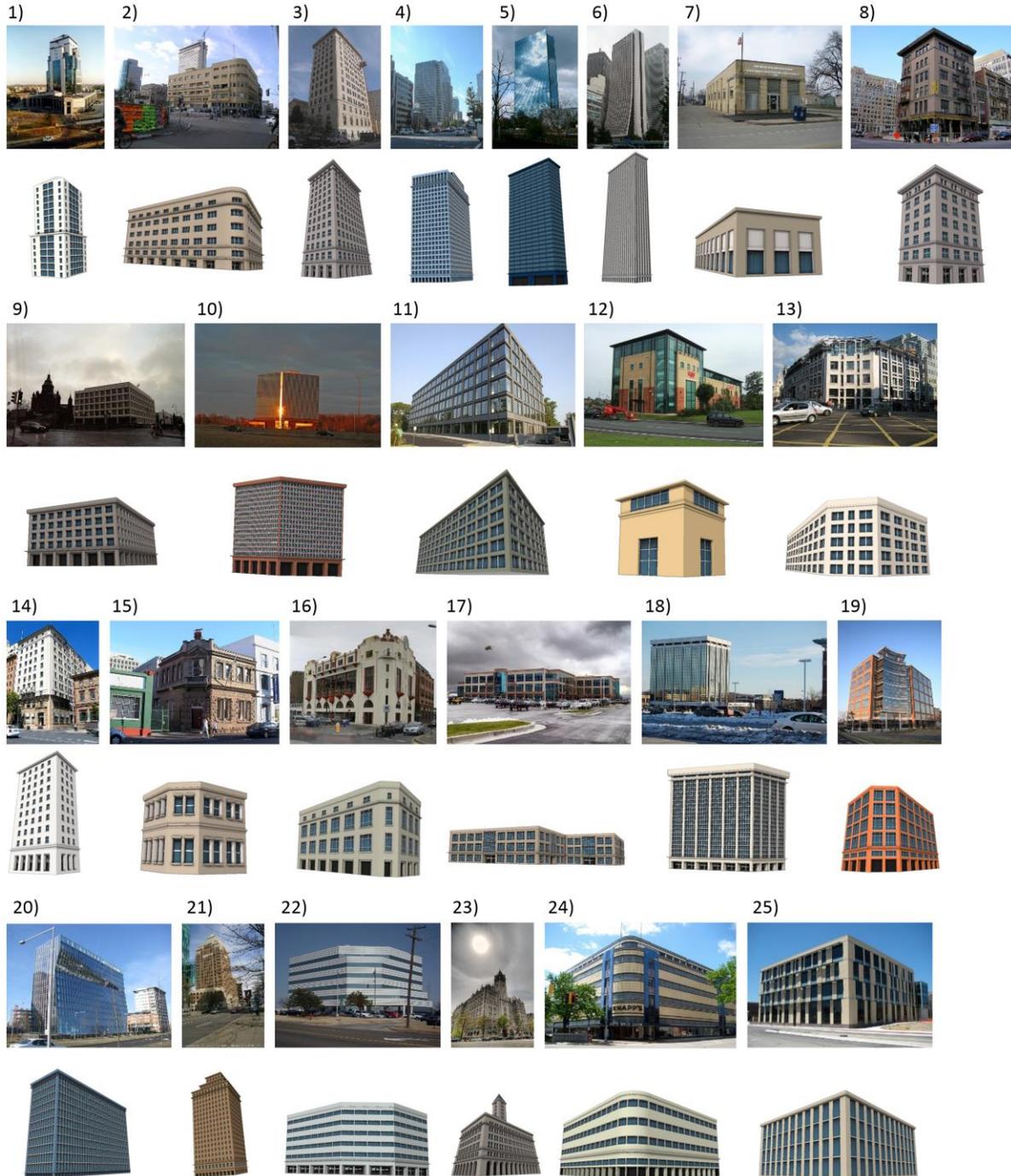


Figure F: The results of our first 25 buildings from ImageNet. For each building, the top image shows the input photo and the bottom image is the output 3D building.

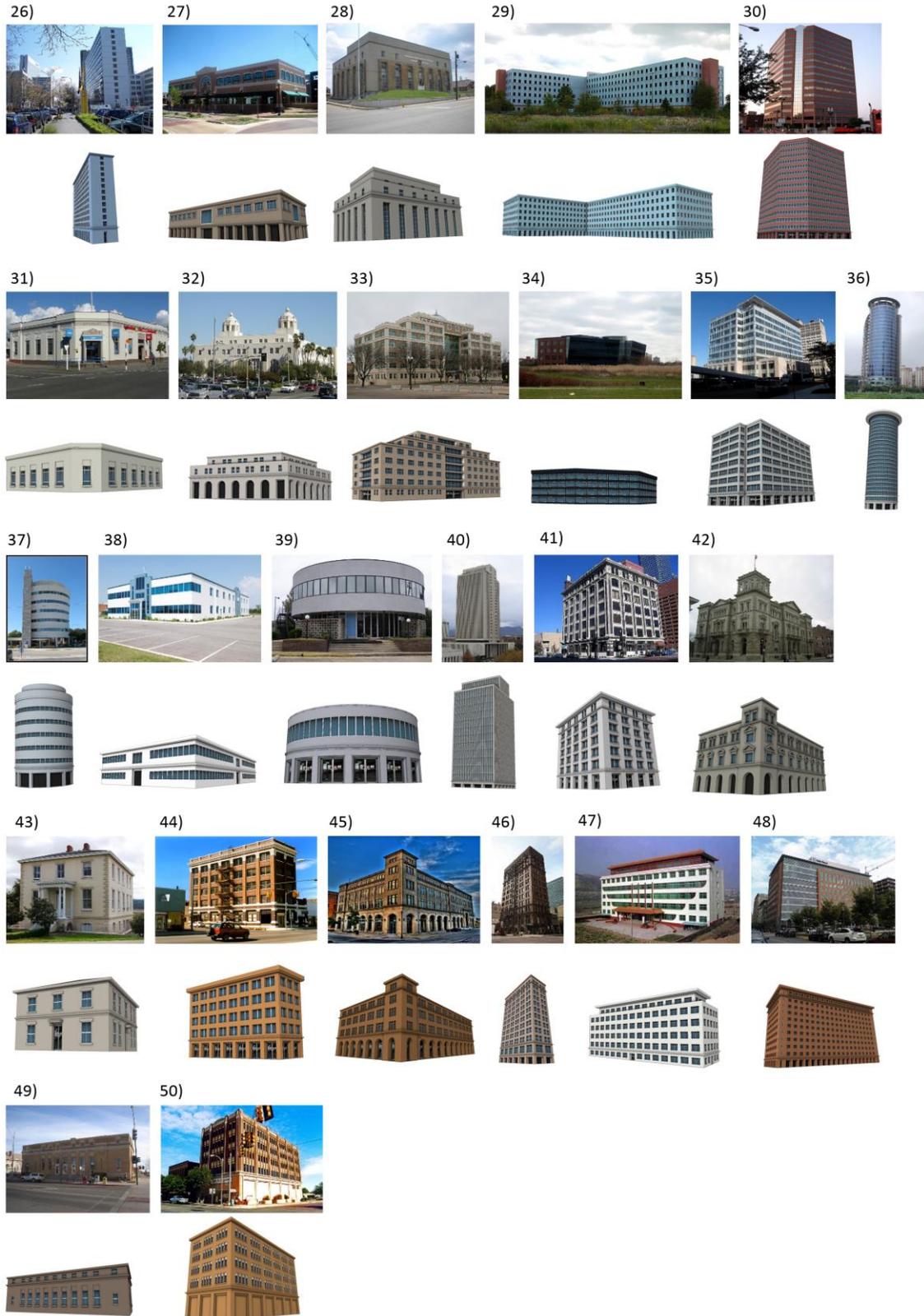


Figure G: The results of our next 25 buildings from ImageNet. For each building, the top image shows the input photo and the bottom image is the output 3D building.

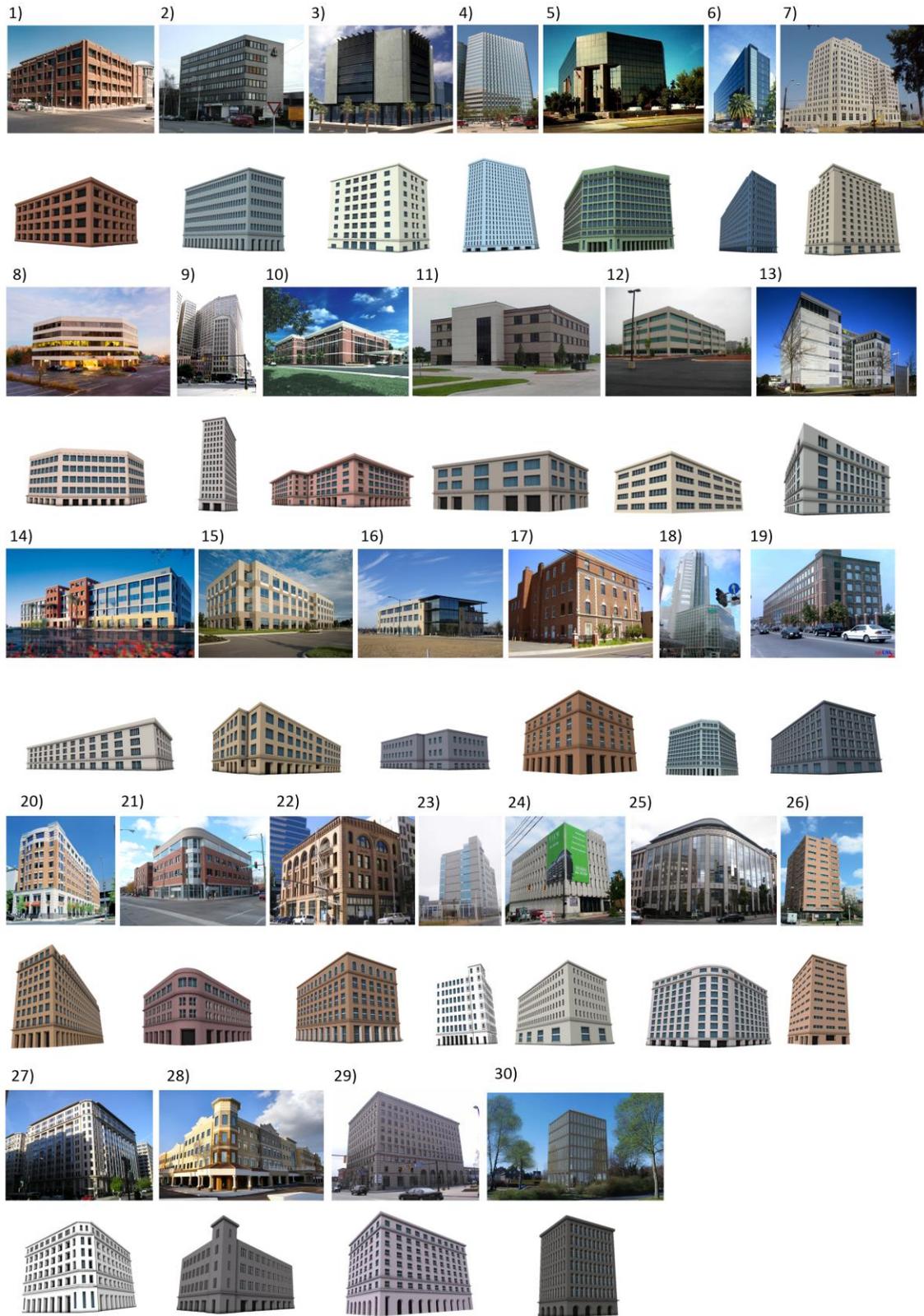


Figure H: The results of our first 30 buildings from SUN dataset. For each building, the top image shows the input photo and the bottom image is the output 3D building.

7 Intermediate Steps

Figure I shows the intermediate steps for some of the results in Figures F, G, and H. The first column shows the input photographs. The second column shows the rectified façade images. The third column shows the parsed façade images as well as the recognized façade grammars. The last column shows the selected window grammars.

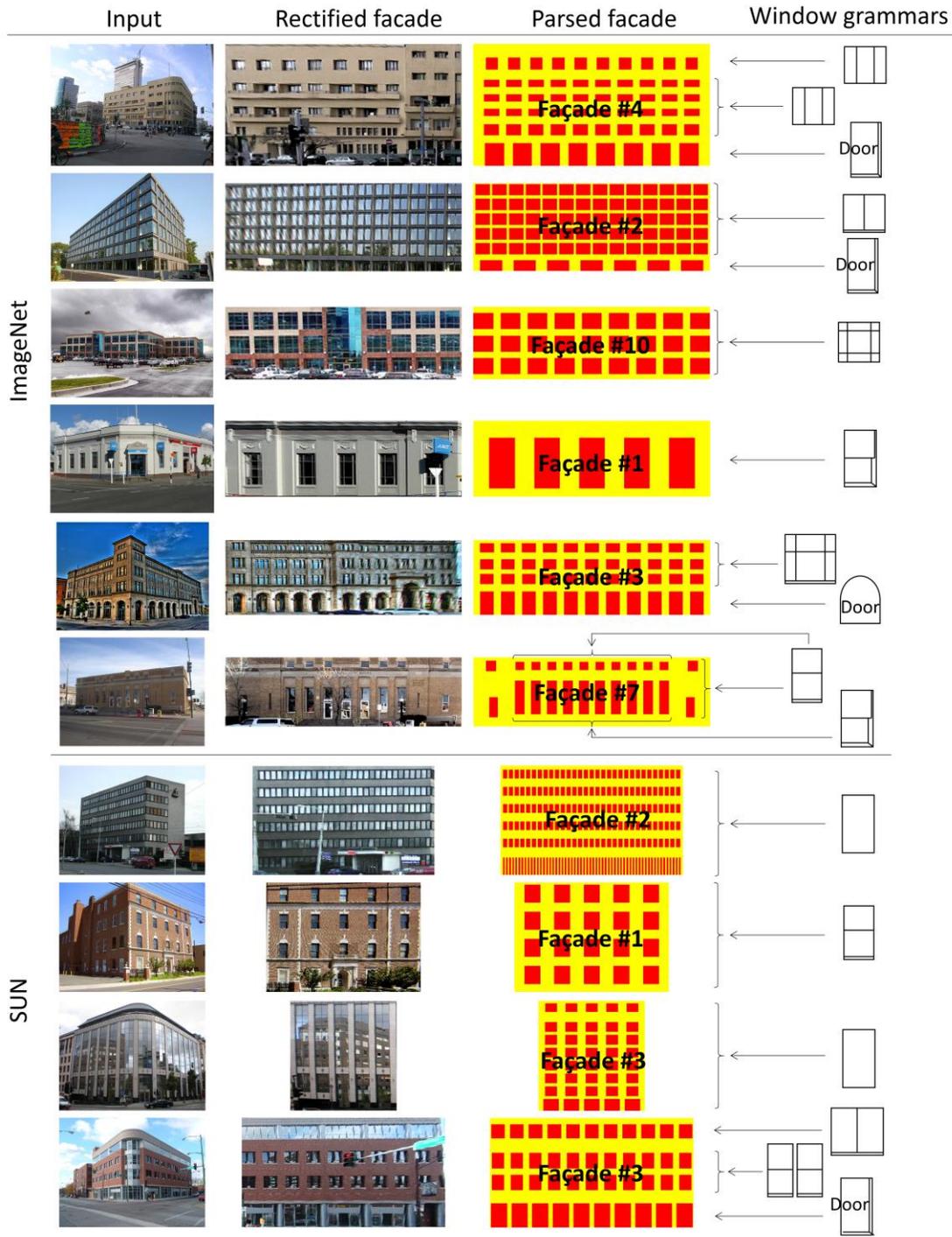


Figure I: Intermediate results of our approach.

8 Façade Parsing Comparison

Figure J shows the façade parsing comparison between the previous approaches and our approach. The previous approaches [Wu et al. 2010; Teboul et al. 2011; U-Net, DeepFacade] do not work well for the office building façade images even though their approaches work well for the ECP dataset. In contrast, our approach works well for both the office building façade images and the ECP dataset.

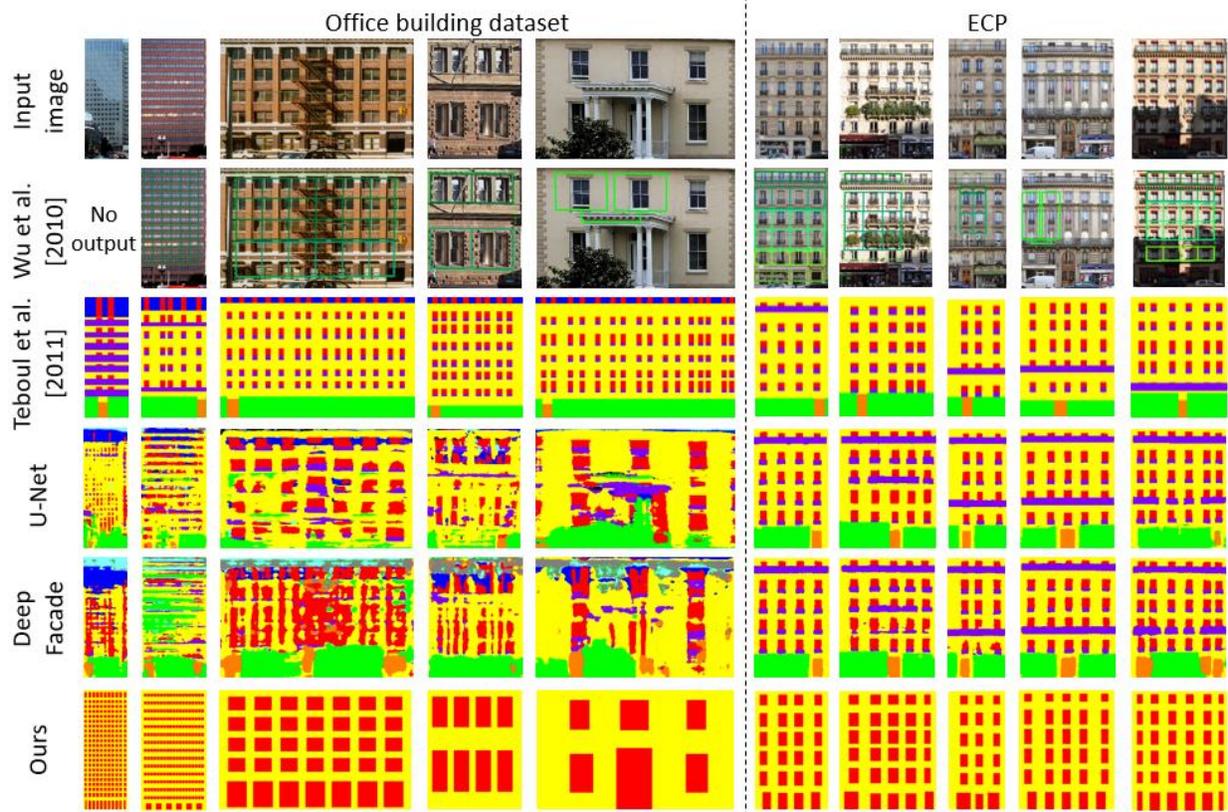


Figure J: Façade parsing comparison.

Table A. Pixel-wise accuracy of façade parsing. For each method, the precision, recall, and F1-score are measured for window and wall classification in addition to the total accuracy.

		Teboul et al. 2011	U-Net	DeepFacade	Ours
ECP	Window	0.74 / 0.73 / 0.73	0.94 / 0.93 / 0.94	0.88 / 0.94 / 0.91	0.56 / 0.60 / 0.58
	Wall	0.89 / 0.89 / 0.89	0.97 / 0.98 / 0.94	0.98 / 0.95 / 0.96	0.83 / 0.81 / 0.82
	Total	0.85	0.96	0.95	0.75
Office building	Window	0.54 / 0.34 / 0.42	0.71 / 0.37 / 0.48	0.66 / 0.49 / 0.56	0.77 / 0.63 / 0.69
	Wall	0.58 / 0.76 / 0.66	0.62 / 0.87 / 0.73	0.64 / 0.78 / 0.71	0.73 / 0.84 / 0.78
	Total	0.57	0.64	0.65	0.74

9 Multi-Color Façade

We considered computing one dominant color for each non-terminal node of the façade grammar. This idea improves the generated façade in some cases, but we found that this local color computation was too sensitive to shadows and occlusions (Figure K).

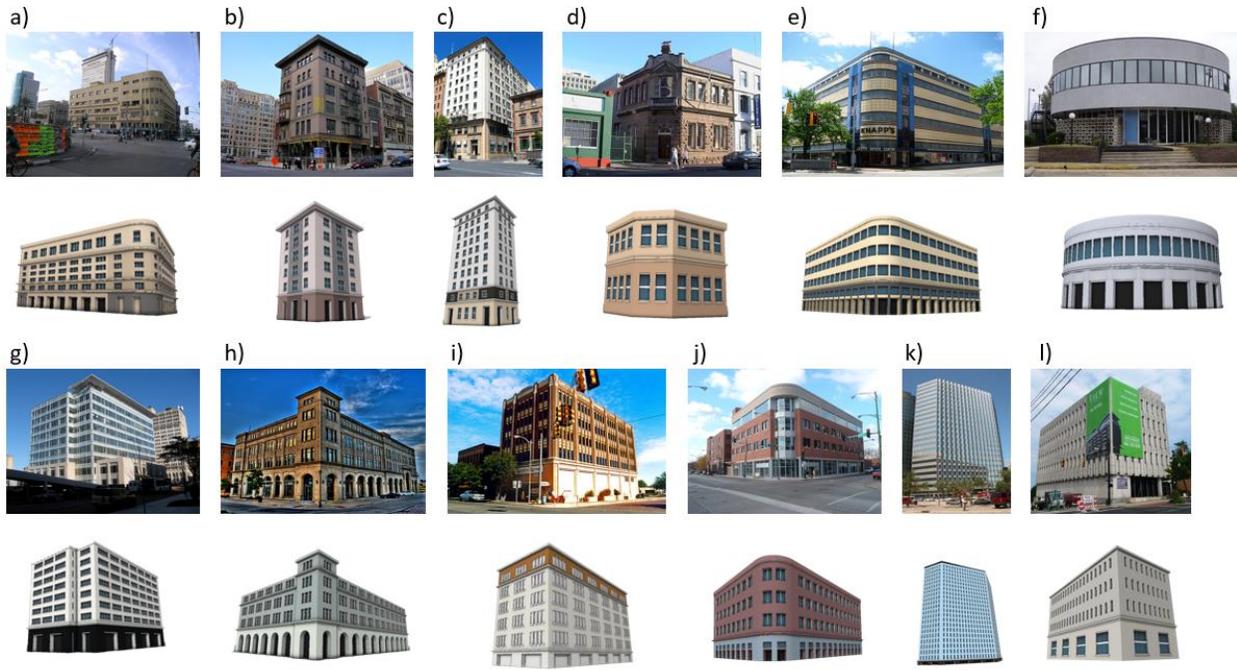


Figure K. The results of using multiple colors for the façade.