Pacific Graphics (2013)
B. Levy, X. Tong, and K. Yin (Editors)

Component-aware Semantic Modeling of Architectures from a Single Image

Fei Hou[†1] Qinping Zhao[1‡] Yue Qi[1§] Hong Qin[2¶]

[1]Beihang University
[2]Stony Brook University

Abstract

This paper advocates a new component-aware framework to reconstruct 3D architecture from a single image. Different from existing work, our motivation is to obtain a complete set of semantically-correct 3D architectural components, which enables part reusability towards rapid model reproduction and facilitate model variation. The core of our system is a novel algorithm to adaptively segment repeated curved stripes (e.g., roof tiles, building floors) into individual elements, based on which 3D dimensions as well as architectural components are derived from a single image. Specially for Chinese architectures, we further devise an interactive method to identify outer columns based on user-specified inner columns. Finally, 3D components are generated and shape rules are derived, from which the buildings and their variants are constructed. Our new component-aware framework emphasizes component utility during rapid 3D architecture reproduction.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1 Introduction and Overview

Rapid creation of 3D architectural models with functionalities is of great significance for interactive 3D graphics, heritage preservation, urban simulation at city-scale, game/film production, etc. Although architectural models can be reconstructed from various data sources, such as 3D scanner data, airborne data and multiple street-view images, such data sources heavily rely on either tremendous manpower or specialized equipment of high cost. Moreover, the conventional image-based modeling methods lack of well-defined semantic and functional contents, which may prevent the model variation and further reuse. We concentrate on structural analysis of Chinese architectures or curved building façades and observe that the curved alternately repeated patterns, are pervasive in both ancient and modern architectures, which afford important clues for 3D dimension recovering and component generation/reuse.

In this paper, we devise a novel unified framework to build a complete set of semantically-correct 3D architectural components as well as 3D models for Chinese and modern architectures. Our motivation is to advance the structural analysis of Chinese architectures and curved façade with fewer interactions. In particular, we detail a novel repetition analysis algorithm to segment curved stripes into individual ones (e.g., roof tiles). After calibrating the camera, for curved modern building façades, we present an algorithm to extract the curved surfaces based on their grid patterns. Specially for Chinese architectures, we present a method to identify the outer columns with the help of user-specified inner columns locations. Based on our originally-developed segmentation and 3D reconstruction, we afford the construction of semantic 3D components and derive the shape rules, from which the 3D architectural models along with their possible variants are generated. Our primary contribution is that we design an alternately repeated pattern analysis algorithm to segment curved stripes adaptively, based on which the 3D dimensions as well as architectural components can be derived

† houfei@vrlab.buaa.edu.cn
‡ zhaoqp@vrlab.buaa.edu.cn
§ qp@vrlab.buaa.edu.cn
¶ qin@cs.stonybrook.edu

DOI: 10.2312/PE.PG.PG2013short.053-058

from a single image. Our secondary contribution is that we develop a complete component-aware semantic architecture modeling system from a single image.

2 Related Work and Our Motivation

We briefly review the most related work in the following categories, which naturally lead to our rationales.

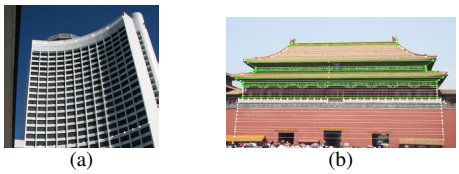
Procedural architecture modeling. Procedural modeling is a powerful method to generate 3D models automatically. Wonka et al. [WWSR03] introduced a powerful split grammar for architectural procedural modeling. Müller et al. [MWH*06], while extending the split grammar, also introduced the CGA shape grammar capable of generating massive urban models. Hou et al. [HQQ12] constructed Chinese architectures procedurally from façade drawing. In addition to the rule-based procedural modeling, Merrell et al. [MM11] presented an example-based model synthesis method. However, these example-based model synthesis methods are difficult to precisely control their results. Instead, we present an inverse procedural modeling approach to extract rules from a single image automatically.

Single view architectural modeling. Modeling from a single image is an ill-posed problem which needs additional clues to recover 3D information. Much work has focused on recovering 3D shapes based on constraints such as symmetry [HYHM04]. Based on symmetry, Jiang et al. [JTC09] presented a framework to build symmetric models, such as Chinese architectures, from a single image. They focused on camera calibration from a frustum but users have to mark all the architectural structures explicitly via interaction.

Repetition analysis. Liu et al. [LHOKG10] and Mitra et al. [MPWC12] presented comprehensive reviews of symmetry analysis in image and geometry, respectively. In the aspect of architectural modeling, Müller et al. [MZWVG07] and Musialski et al. [MWW12] detected symmetric or partially symmetric elements in façade image to reconstruct façade procedurally. Shen et al. [SHFH11] presented an adaptive urban façade partition algorithm. However, in order to split irregular façades, this algorithm does not exploit regular repetitions sufficiently in the first splitting, which would make it unstable. Our strategy is rather different. We exploit irregular repetitions based on regular repetitions.

3 Initialization by User Interaction

Given an image, the user draws the façade contour (Fig. 1). The vertical (modern building) or horizontal (Chinese architectures) vanishing point is computed as the intersection of the two vertical (horizontal) segments of the contour. For ancient Chinese architectures, as shown in Fig. 1b, the user should also provide an initial vertical segmentation (denoted as green lines). The system first computes some horizontal candidate lines passing through the vanishing point with larger gradients and then the user selects necessary lines from candidates or draw additional lines to segment the façade into semantic partitions and identify their types, such as ridge, roof, bracket set lintel, room, platform, etc.



(a) (b)

Figure 1: Interactive façade segmentation. (a) A hotel model. The user first draws the façade contour. (b) Meridian Gate. The user specifies necessary lines (colored in green) to segment the façade vertically, where the white lines are candidates with larger gradients.

4 Segmentation of Alternately Repeated Stripes

The common characteristic of roof tiles (Fig. 4) of Chinese architectures and floors (Fig. 5b) of modern buildings is that, they essentially present curved stripe shapes and alternately repeated patterns, i.e., a repeated sequence of $\{a,b,a,b,a,b,\ldots\}$. The roof tiles are alternately repeated regularly, i.e, the repeated stripes cover the whole surface, while some building floors, such as Fig. 9a, are alternatively repeated irregularly due to the existence of the purple dashed regions, i.e., they cover part of the surface. We now present an automatic algorithm to segment the regular or irregular curved stripes into individual ones. We segment the stripes (Section 4.2) based on the stripe directions (Section 4.1).

4.1 Stripe Direction Field

We first evaluate the stripe directions which are a prerequisite for stripe segmentation.

Evaluating direction at a point. Although the stripes are curved, they are nearly straight locally. We regard the stripes as several alternately repeated straight lines locally. We observe that the tiles are locally bilateral symmetric with respect to the axis orthogonal to its direction. We find the direction of this axis by iteratively enumerating from 0° to 180° at the pixel to minimize the intensity difference between corresponding symmetric pixels.

Evaluating directions on the surface. As shown in Fig. 2, we evenly sample several (10 in the example) horizontal lines (*sample line*) passing through the horizontal vanishing point (e.g., the white lines). In 3D space, the tangent directions are parallel on each sample line. Therefore in 2D image space, the tile directions on each sample line should be intersecting at a vanishing point and the vanishing points on different sample lines should be collinear. And for each sample line, after computing the tile directions at several (50 in the example) sample points, we vote for their corresponding vanishing points using RANSAC followed by computing the vanishing line using RANSAC, where the incorrect vanishing points are filtered out. The result is shown in Fig. 2.

4.2 Adaptive Stripe Segmentation

We devise a heuristic algorithm to adaptively segment the regular (irregular) repeated stripes into individual ones. Whether it is regular or not is specified by the user. The

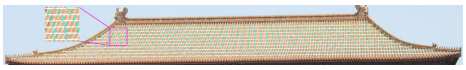


Figure 2: The tile direction field. The lines (colored in blue) indicate the tile directions at the sample points on the sample horizontal lines (colored in white).

critical idea behind this algorithm is that, we evaluate the repeated patterns hierarchically in different scales and then combine them to vote for the best split sequences to split the stripes. We extract candidate splitter subsequences in Sec. 4.2.1, and then combine them on every sample lines (Sec. 4.2.2) and the whole surface (Sec. 4.2.3), respectively.

On each sample line, we sample an ordered sequence of points $S = \{p_1, p_2, \ldots, p_n\}$ as splitter candidates with spacing of one pixel. Given a subsequence $S' \subset S$, $S' = \{p'_1, p'_2, \ldots, p'_n\}$, the set of continuous alternately repeated subsequences of S' is denoted by $\mathcal{R}(S')$. A subsequence $l \in \mathcal{R}(S')$, if the following conditions hold:

1. (*continuous*) $l = \{p'_k, p'_{k+1}, \ldots, p'_{k+m}\}$.
2. (*2-alternately repeated*) $\min(\|p'_i - p'_{i+1}\|, \|p'_{i+2} - p'_{i+3}\|)/\max(\|p'_i - p'_{i+1}\|, \|p'_{i+2} - p'_{i+3}\|) \geq \tau_1, k \leq i \leq k+m-3$. In the irregularly repeated case, the textures between intervals (p'_i, p'_{i+1}) and (p'_{i+2}, p'_{i+3}) should also be similar.
3. (*longest*) Neither the subsequence $\{p'_{k-1}, p'_k, p'_{k+1}, \ldots, p'_{k+m}\}$ nor $\{p'_k, p'_{k+1}, \ldots, p'_{k+m+1}\}$ satisfies condition 2.

4.2.1 $\mathcal{R}(S)$ Extraction on a Line

Given candidate splitters S, we have observed that a point with larger derivative with respect to the direction cross the stripes (i.e. orthogonal to the stripe direction) is more likely to be a splitter. For every candidate p_i, we denote its potential to be a splitter by $\mathcal{P}(p_i)$ and evaluate it as follows: let the sum of differences between symmetric pixels with respect to the stripe direction at p_i be $d(p_i)$. $\mathcal{P}(p_i) = r$, *iff.* in the neighborhood $\{p_{i-r}, \ldots, p_i, \ldots, p_{i+r}\}$, $d(p_i) > d(p_k)$, $k \neq i, i-r \leq k \leq i+r$, and $d(p_i) < d(p_{i-r-1})$ or $d(p_i) < d(p_{i+r+1})$. We analyze the points $p_i \in S$ hierarchically based on their potentials to reveal all possible potential repeated patterns contained in it. The sample points in S are classified into different levels of subsequences $S^r = \{p_i\}$ where $p_i \in S^r$ *iff.* $\mathcal{P}(p_i) \geq r$.

Definition 1: Suppose l_1 and l_2 are two sequences. If the union sequence of l_1 and l_2 is also 2-alternately repeated, l_1 and l_2 are called *merge-able*.

Given S, we extract $\mathcal{R}(S)$ repeatedly by varying r from maximum to 1. In each loop, first, for every element $l_i \in \mathcal{R}(S)$, we expand l_i by S^r. If a subsequence $s \in \mathcal{R}(l'_i)$ satisfies $l_i \subset s$, then l_i is substituted by s. Second, we compute $\mathcal{R}(S^r)$, and for every subsequence $s \in \mathcal{R}(S^r)$, if no $l_i \in \mathcal{R}(S)$ s.t. $l_i \supseteq s$, then s is inserted into $\mathcal{R}(S)$. After a few iterations, $\mathcal{R}(S)$ will contain all the potential splitter subsequences which might be heavily overlapped. Third, we merge all the subsequences in $\mathcal{R}(S)$ that are merge-able.

4.2.2 Stripe Segmentation on a Line

The subsequences in $\mathcal{R}(S)$ are redundant and heavily overlapped with different repeated patterns. Next, we try to extract a set of consistent and disjoint subsequence groups to segment the stripes on every sample line independently. To ease the discussion, we first define some notations.

Definition 2: Let l_1 and l_2 be two alternately repeated subsequences. If l_1 and l_2 are not merge-able and the spacings and textures between l_1 and l_2 are both similar, l_1 and l_2 are called *compatible* (i.e. l_1 and l_2 are with the same repeated pattern). Let G_1 and G_2 be two sets of subsequences. If all pairs of $l_i \in G_1$ and $l_k \in G_2$ is either merge-able or compatible, G_1 and G_2 are called *combinable* (i.e. G_1 and G_2 belong to the same group). If G_1 and G_2 are not combinable and all pairs of $l_i \in G_1$ and $l_k \in G_2$ is not overlapped, G_1 and G_2 are called *group-able* (i.e. G_1 and G_2 are not mutual exclusion).

The subsequences in $\mathcal{R}$ are first classified into different groups according to their compatibility (grouping step), and second, certain groups are extracted and combined to split the line (combination step).

In the grouping step, the subsequences in $l_i \in \mathcal{R}(S)$ are sorted in descending order according $len(l_i) \times num(l_i)$, where $len(l_i)$ is the distance between the first and last splitters of l_i and $num(l_i)$ is the number of splitters in l_i. The subsequences in $\mathcal{R}(S)$ are processed in order to classify them into different groups $\mathcal{G} = \{G_i\}$, satisfying $\bigcup_i G_i = \mathcal{R}(S)$. If $l_i, l_j \in G_k$, l_i and l_j must be compatible. This procedure is repeated until all $l_i \in \mathcal{R}(S)$ are processed. In the combination step, all $G_i \in \mathcal{G}$ are sorted in descending order according to $(\sum_{l_k \in G_i} len(l_k) \times \sum_{l_k \in G_i} num(l_k))/\#(G_i)$, where $\#(G_i)$ is the number of sequences in G_i. They are processed in order, to extract disjoint and compatible groups, whose set is denoted by $\mathcal{L}$. Given a group $G_k \in \mathcal{G}$, if G_k is combinable with some $G_i \in \mathcal{L}$ and group-able with all the other $G_i \in \mathcal{L}$, G_k is merged with all the combinable groups. If G is not combinable with any $G_i \in \mathcal{L}$ but group-able with all the $G_i \in \mathcal{L}$, G_k is inserted into $\mathcal{L}$. Otherwise G is discarded. This procedure is repeated until all $G_k \in \mathcal{G}$ are processed.

Figure 3: Stripe segmentation on a line. Different groups of splitters are drawn in different colors.

4.2.3 Stripe Segmentation on Curved Surface

After splitting on each sample line independently, some lines may not split appropriately (Fig. 3). In this step, we combine the splitters on all the sample lines to vote for the most appropriate splitters to split the stripes eventually. Every splitter is traced along the stripe direction field to its corresponding position on a common base line (always the middle one).

Similar to the combination step on a single line, we also need to combine these groups to split the base line except that we tolerate certain deviations between corresponding splitters while merging, and average their positions since errors are introduced during splitter tracing along stripe directions. Then, we complete the gaps between the subsequences of $\mathcal{L}$ according to the splitters near the gaps. If it is irregular splitting, the gaps near the splitters with similar textures to the segmented stripes are filled. Finally, the groups except G_0 are discarded as non-repeated region (e.g., dashed regions in Fig 9a). If it is regular splitting, we force all the gaps are filled according to the repeated pattern of G_0. Finally, from these splitters, the complete stripes are traced along the stripe direction field to segment the entire surfaces as shown in Fig. 4 and Fig. 5b (the top façade is split separately), respectively.

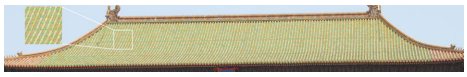
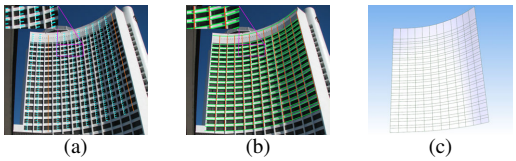


Figure 4: Regular segmentation for roof tile.



(a) (b) (c)

Figure 5: (a) The floor direction field. (b) Segmentation on sample lines. (c) The irregular segmentation for hotel building façade. (d) The reconstructed grid mesh.

5 3D Reconstruction

Geometry reconstruction from a single image is an ill-posed problem needing additional clues. We assume the intrinsic parameters only have one degree of freedom (i.e., focal length) and fit the focal length based on the orthogonality of parallel stripes and the horizontal (vertical) direction. Later, we develop two 3D reconstruction methods for ancient Chinese architectures and curved modern façades, respectively.

5.1 Ancient Chinese Architecture Reconstruction

Due to the symmetry, similar to Jiang [JTC09], we flip the image horizontally to get a second virtual camera which mimics the image captured at the symmetric position. We sweep the horizontal lines intersecting the contour at two points p_1 and p_2 and we obtain two pairs of correspondences (p_1, p_2') and (p_2, p_1'), where p_1' and p_2' are the horizontally flipped p_1 and p_2 respectively. Then, based on these intersections, we evaluate the projection matrices of the two cameras followed by bundle adjustment. Finally, we get the projective matrices and the 3D contour of the building.

5.2 Curved Modern Building Façade Reconstruction

Different from Chinese architectures, making use of the grid pattern, we present a novel method to recover the 3D curved façade. First, the geometry is recovered by developing rectangle patches successively. Second, the geometry and camera are refined in a non-linear least square procedure.

As shown in Fig. 5, the parallel curved stripes are piecewise linear and consisting of rectangular patches, so given the intrinsic parameters, we can rotate the camera to rectify the rectangular patch successively by a homography to rectify the patches. Thus, the initial 3D coordinates $\{V_{ij}\}_{i,j=0}^{m,n}$ of the corners (i.e., grid points) are derived, where i denotes the row (floor) index and j denotes the column index.

We initialize camera projection matrix $\mathcal{P}$ according to the 3D points $\{V_{ij}\}_{i,j=0}^{m,n}$ and their 2D projections $\{p_{ij}\}_{i,j=0}^{m,n}$. If the building is symmetric (which is specified by the user), we use principal component analysis to compute its symmetric plane P_{sym} and minimize Eq. 1 to optimize $\{V_{ij}\}_{i,j=0}^{m,n}$, $\mathcal{P}$ and P_{sym}. In order to enforce the symmetry, for every $V_{i_0 j_0} \in \{V_{ij}\}_{i,j=0}^{m,n}$, we generate a ghost point $V'_{i_0 j_0}$ to represent its symmetric point, which is initialized to the point on the 3D floor curve $\{V_{i_0 j}\}_{j=0}^{n}$ (i.e., the $V_{i_0 j_0}$'s row of the grid points) closest to the symmetric point of $V_{i_0 j_0}$.

$$\begin{aligned} d =& \alpha \sum_{i,j} d_1^2(\mathcal{P}V_{ij}, p_{ij}) + \beta \sum_{i,j} d_2^2((V_{ij}+V'_{ij})/2) \\ &+ \gamma \sum_{i,j} \theta^2(V_{ij}-V'_{ij}) + \delta \sum_{i,j} d_3^2(V'_{ij}), \end{aligned} \tag{1}$$

where $d_1(\mathcal{P}V_{ij}, p_{ij})$ is the Euclidean distance from the projection of V_i to its corresponding image point p_{ij}, and $d_2((V_{ij}+V'_{ij})/2)$ is the distance from point $(V_{ij}+V'_{ij})/2$ to plane P_{sym}. $\theta(V_{ij}-V'_{ij})$ is the sine of the angle between vector $V_{ij}-V'_{ij}$ and the normal of P_{sym}. $d_3(V'_{ij})$ is the minimum distance from point V'_{ij} to the 3D floor curve $\{V_{ij}\}_{j=0}^{n}$. In the case of non-symmetric building, only the first term is retained, which degenerates to ordinary bundle adjustment. We use Levenberg-Marquardt method to minimize Eq. 1 iteratively. In each iteration, the variables $\{V_{ij}, V'_{ij}\}_{i,j=0}^{m,n}$, P_{sym} and $\mathcal{P}$ are optimized independently in turn, while the others are fixed. In Eq. 1, the points on the same floor are enforced to lie on a common horizontal plane and the points vertically aligned are enforced to lie in a common vertical line. In our experiments, we set $\alpha = 4$, $\beta = 1$, $\gamma = 10$ and $\delta = 1$. Fig. 5c shows the recovered 3D grid mesh of the building in Fig. 5b.

6 Column Segmentation

For Chinese architectures, the columns divide the room into bays (the areas between adjacent columns). We have to identify the columns and thus the bays and lintels are derived. For architectures with corridors, the inner bays are occluded by outer columns, we present an interactive method to identify outer columns based on user-specified inner columns.

Initialization. The user first identifies the locations of the

inner columns and draws the width of an arbitrary one. If the two side columns are not at the two ends, they are moved to the two ends for outer column segmentation. We sample a horizontal line at the middle height of the bays, to which the user-specified points are mapped along vertical direction and denoted by $\{c_0,\ldots,c_m\}$, and recover their 3D coordinates $\{C_0,\ldots,C_m\}$ by symmetry. Also, we fit the outer column plane P_{out} based on 3D points on the contour, and the inner column plane P_{in} is fitted on $\{C_0,\ldots,C_m\}$ constrained by the normal of P_{out}. For buildings without corridors, we draw the outer columns and the system determines whether the corridor exists automatically based on the ratio of the distance between the two fitted planes and the room height. Finally, $\{C_0,\ldots,C_m\}$ are projected onto P_{out} to get $\{C'_0,\ldots,C'_m\}$ as the initial positions of outer columns (Fig. 6), which are further refined in the next step.

Figure 6: Column segmentation. The yellow points are the initial locations of outer columns. After refinement, the final inner columns and outer columns are marked by magenta lines and white line respectively.

Iterative refinement. The bays are always evenly distributed except for the center and/or two side ones. We first refine $\{C'_0,\ldots,C'_m\}$ based on the symmetry and uniformity. $\{C'_0,\ldots,C'_m\}$ are first parameterized by a scalar on their line, but we still use these symbols and minimize,

$$\{C'_i\}_{i=1}^m = \underset{\{C'_i\},1\le i\le m}{\arg\min}\ (\alpha \sum_{uniform\ bays} \|C'_i + C'_{i+2} - 2C'_{i+1}\|^2) + \beta \sum_{i<m/2} \|(C'_i + C'_{m-i}) - (C'_{i+1} + C'_{m-(i+1)})\|^2) + \sum_{1\le i\le m} \|C'_i - C'^0_i\|, \quad (2)$$

where the first term constrains the uniformities, and the second term constrains their symmetry. C'^0_i is the initialized column locations. We repeatedly refine $\{C'_i\}_{i=1}^m$ by above equation until the variations of C'_i are small enough or exceed maximum number of iterations. After refinement, $\{C'_0,\ldots,C'_m\}$ are projected onto the image to get $\{c'_0,\ldots,c'_m\}$. Since the inner and outer columns are aligned, the mapping between $\{c_0,\ldots,c_m\}$ and $\{c'_0,\ldots,c'_m\}$ is subject to a 1D homography. The columns are much smoother than the bays since the windows or doors are always decorated, meaning that the gradients on columns are smaller than that of bays. We iteratively refine the outer column positions based on the homography and image gradient alternately. Finally, the vertical lines with larger gradient are reserved as the candidate splitting lines and the iteratively refined columns and the intervals divided by the candidate splitting lines are combined to segment the columns. The identified columns are shown in Fig. 6. The lintels are aligned with the columns in 3D, so they are segmented passingly by projecting the columns onto their planes.

7 Component, Rule, and Model Generation

It now sets the stage to construct semantic 3D architectural components, which are the basic elements to compose the 3D models reusable for various model generation.

Chinese architecture components. Some of the components of the Meridian Gate are shown in Fig. 7a. To complete the occluded textures of bays, making use of the symmetry of the bay, we flip the bay horizontally and register them followed by the use of graph cut texture synthesis [KSE*03] to stitch the two images. Finally, the repaired image is textured on a planar surface to form the bay component. The column and lintel column are simply formulated as a cylinder textured by their images, and the lintels are formulated as planar surfaces.

(a) (b)

Figure 7: (a) A subset of the 3D components of the Meridian Gate. (b) Some of the components of the hotel model.

Modern architecture components. After segmenting the curved façade into grid pattern as shown in Fig. 5b, every rectangular patch forms a rectangle component textured by the image (Fig. 7b).

Rules Derivation. We integrate the image-based modeling method with the procedural modeling method. Finally, we derive GGA [MWH*06] shape grammar rules automatically to compose these 3D components in order to construct semantic architectures and their variants. The rules construct mass (coarse) models at first, whose depth differences equal to the width differences to the roof and then split a top-down manner to combine the architectural components. The generated Meridian Gate and one of its variants are shown in Fig. 8a and Fig. 8b, respectively. The final hotel model is shown in Fig. 8c and a stretched building is shown in Fig. 8d, where the number of windows and walls grows adaptively.

(a) (b) (c) (d)

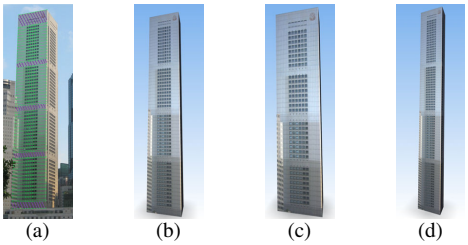
Figure 8: (a) The Meridian Gate model. (b) The stretched model. (c) The hotel model. (d) The stretched model.

8 Experimental Results and Evaluations

We have compared the Meridian Gate model with the ground truth whose width (the room partition) is 60.05m and depth is 25.00m. The relative depth error of the result, whose width is 1.733 and depth is 0.754 (26.13m), is about 4.5%. Please refer to the affiliated video for more results.

Except for curved façades, the planar façade building UOB Plaza is shown in Fig. 9a, which is irregularly split.

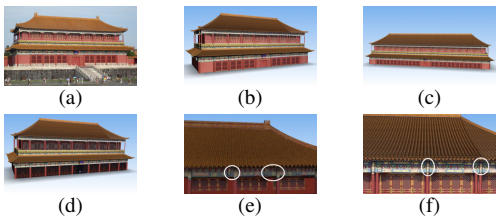
The window floors are classified into a repeated group G while the dashed areas are split separately. The reconstructed building is shown in Fig. 9b and a shrunk building is shown in Fig. 9c, where the floors are generated adaptively. Fig. 9d shows a stretched building where the floors are adjusted adaptively.



(a) (b) (c) (d)

Figure 9: The UOB Plaza. (a) The building is adaptively split into repeated floors, which is irregular due to the purple dashed regions. (b) The reconstructed UOB building. (c) The shrunk building. (d) The stretched building.

Comparison. Fig. 10b shows our reconstructed Pavilion of Manifest Benevolence (also known as TiRenGe) from the input image (Fig. 10a). Jiang et al. [JTC09] also has reconstructed the same model. Their method, however, needs to draw a frustum, roof curves, roof tiles, walls, columns, and others, for calibration and building generation. Nonetheless, they did not explicitly construct architectural components, so their building is only a static model. In contrast, our building can be extended adaptively (Fig. 10c) and its components can be substituted because of our component-aware modeling capability. The bays can be substituted as shown in Fig. 10d. The lintels of Jiang et al.'s method have exhibited mis-alignment (highlighted inside circles, Fig. 10e) because they are never segmented into individual components. Furthermore, the inner columns are not reconstructed either. In contrast, we construct the appropriate columns and lintels as shown in Fig. 10f. Our automatic tile segmentation algorithm produces more accurate tile density compared with Jiang's manual segmentation.



(a) (b) (c)

(d) (e) (f)

Figure 10: (a) The input TiRenGe image. (b) Our reconstructed model. (c) Our extended model. (d) The deformed model with different bays. (e) A local close-up view of Jiang's model [JTC09], the mis-alignment is highlighted (Thanks to Ping Tan for providing their final model). (f) The same localized view of our model for comparison.

9 Conclusion and Future Work

We have detailed a component-aware semantic modeling framework to reconstruct architectures from a single image. The core of the system is an adaptive curved stripe segmentation algorithm to segment building tiles and floors, based on which the 3D dimensions are recovered from a single image. Finally, the 3D semantic components as well as shape rules are generated, from which the 3D models and their variants are constructed. In the future, to reduce users' burden for costly and extensive interaction, the building contour could be segmented by the image matting approach and the building partitions could be recognized by using computer vision techniques.

Acknowledgement: This work is supported in part by National Natural Science Foundation of China (No. 61300068, 61190120, 61190121, 61190125, 61073078, 61272348 and 61202235), National Science Foundation of USA (No. IIS-0949467, IIS-1047715, and IIS-1049448), PostDoc Research Funding (No. 2013M530512) and Ph.D. Program Foundation of Ministry of Education of China (No. 20111102110018).

References

[HQQ12] HOU F., QI Y., QIN H.: Drawing-based procedural modeling of chinese architectures. *IEEE Transactions on Visualization and Computer Graphics 18*, 1 (2012), 30–42. 2

[HYHM04] HONG W., YANG A. Y., HUANG K., MA Y.: On symmetry and multiple-view geometry: Structure, pose, and calibration from a single image. *Int. J. Comput. Vision 60* (2004), 241–265. 2

[JTC09] JIANG N., TAN P., CHEONG L.-F.: Symmetric architecture modeling with a single image. *ACM Trans. Graph. 28*, 5 (2009), 1–8. 2, 4, 6

[KSE*03] KWATRA V., SCHÖDL A., ESSA I., TURK G., BOBICK A.: Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph. 22* (2003), 277–286. 5

[LHOKG10] LIU Y., HEL-OR H., KAPLAN C. S., GOOL L. J. V.: Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Computer Graphics and Vision 5*, 1-2 (2010), 1–195. 2

[MM11] MERRELL P., MANOCHA D.: Model synthesis: A general procedural modeling algorithm. *IEEE Transactions on Visualization and Computer Graphics 17*, 6 (2011), 715 –728. 2

[MPWC12] MITRA N. J., PAULY M., WAND M., CEYLAN D.: Symmetry in 3d geometry: Extraction and applications. In *EUROGRAPHICS State-of-the-art Report* (2012). 2

[MWH*06] MÜLLER P., WONKA P., HAEGLER S., ULMER A., VAN GOOL L.: Procedural modeling of buildings. *ACM Trans. Graph. 25*, 3 (2006), 614–623. 2, 5

[MWW12] MUSIALSKI P., WIMMER M., WONKA P.: Interactive coherence-based façade modeling. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2012) 31*, 2 (2012). 2

[MZWVG07] MÜLLER P., ZENG G., WONKA P., VAN GOOL L.: Image-based procedural modeling of facades. *ACM Trans. Graph. 26*, 3 (2007), 85. 2

[SHFH11] SHEN C.-H., HUANG S.-S., FU H., HU S.-M.: Adaptive partitioning of urban facades. *ACM Trans. Graph. 30*, 6 (2011), 184:1–184:10. 2

[WWSR03] WONKA P., WIMMER M., SILLION F., RIBARSKY W.: Instant architecture. *ACM Trans. Graph. 22*, 3 (2003), 669–677. 2