International Journal of Computer Vision and Signal Processing, 6(1), 33-46(2016) ORIGINAL ARTICLE

Constrained Procedural Modeling of Real Buildings from Single Facade Layout

Divya Udayan J*

School of Information Technology and Engineering
VIT University, India

HyungSeok Kim

Internet and Multimedia Engineering
Konkuk University, S.Korea

Abstract

We present a novel modeling framework to generate real-world 3D buildings from a single facade layout that adapt to the real footprint data automatically. Facade components are extracted from the facade layout and organized as a repetitive shape tree. Building facade layout encodes expected architectural constraints and is able to derive complex instances using shape grammars. We extend the previous approaches of procedural building models to a constraint-based framework for the recovery of the hidden parts of the building. We then provide an interactive editing process for updating of the structural topology given a different view of the building. We demonstrate our framework on several real-world buildings with varying amount of complexity in appearance and footprint shape and we show that our approach can generate similar buildings to the original that are used to populate a virtual city depicting implicit culture and style of different ages, while opening new perspectives for image understanding and 3D modeling.

Keywords: architectural image based modeling, single view reconstruction, vision and scene understanding, shape grammars, procedural modeling

ISSN: 2186-1390 (Online)
http://cennser.org/IJCVSP

Article History:
Received: 31 January 2016
Revised: 1 December 2016
Accepted: 24 December 2016
Published Online: 27 December 2016

1. INTRODUCTION

Large scale architectural 3D building modeling has been receiving increasing demand in the fields of urban design, navigation, 3D games and entertainment. It often involves time consuming and laborious manual processing in order to achieve high quality mass modeling. To address this problem, previous works utilize procedural modeling to help generate complex and repetitive structures. However, expressing the designers intent by manually writing the grammar is virtually impossible and that is why many interactive, or user assisted approaches have been used.

Real-world buildings have their unique structures with complex footprints but their architectural components are repetitive along the horizontal and vertical directions. Inspired by this observation and similar to the previous work, we use an user-assisted approach to encode the facade layout as a repetitive component tree. Our pipeline starts from a single structure analysis to extract the shape information given as input a single non-calibrated 2D image. We then, cluster similar components together in a bottom-up manner in order to label the facade into semantically meaningful architectural components. This step gives sufficient information to generate rules for individual component configurations. Rules for the structural topology are derived based on the hierarchically arranged component tree and geospatial information.

The key observation of our work is that component identification, grouping and solving for individual component fitting in a hierarchical and layered structural representation, leads to an enhanced accuracy and editing capability. Although a fully automated solutions for facade encoding exist, to the best of our knowledge, no framework allows for generation of a complete building from the facade and the building footprint. In our work, we propose

---

*Corresponding author
Email address: `divya.udayan@vit.ac.in` (Divya Udayan J)

an interactive user-assisted solution. Beyond the 3D building reconstruction, our framework reveals the possibilities for a layered 3D facade generation, in which signboards and other functional metadata can be placed at the exact positions bringing a real life-like appearance to buildings. In addition, real footprint of the building is given as prior information in order to align the reconstructed facade geometry so that it reflects the real world. The height information is obtained either from real measurement or estimated measurement calculated from the ratio of pixels occupied by the horizontal footprint segment and the pixels occupied by the vertical height of the building image.

In this paper, we focus on the problem of rapid generation of a city based on constrained procedural modeling that adapts to the real footprint data. In this work, we propose a semi-automatic framework to rapidly reconstruct a 3D building model that exactly matches the building photograph taken from a single view using limited prior information. We demonstrate our framework on a range of real-world buildings with varying amount of complexity in appearance and footprint shape. We experimented with the novel synthesis and manipulation possibilities, using single photograph and building footprint as input and in updating pre-constructed 3D models, towards the goal of meaningful and interactive 3D model editing. The main contributions of this work are:

- a rapid reconstruction of exactly matching building with its single view photograph using limited information.
- automatic building adaptation to real footprint data.
- hidden parts recovery and modeling that adapt to the multi-view of the building.
- interactive editing, enabling facade-syntax preserving manipulations resulting in mass modeling of a virtual city with a single family of buildings with variations.
- updating an existing virtual city with different style of buildings depicting implicit culture and style of different ages could be modeled.

The rest of the paper is organized as follows. In section 2, a brief review of the related works is presented. A detailed description of the proposed methodology is given in section 3. In section 4 we present the results and evaluations of the experiments conducted on several real world buildings.

2. RELATED WORK

Our work relates to image-based modeling, symmetry detection, and inverse procedural modeling that we review below.

Image based architectural modeling using single or multiple images to derive formal rules that describe the structural information of buildings are increasing due to easy availability of photographs and recent advances in computer vision. Debevec et al.[1] used manual selection of features and correspondences from different side views to model facades. Oh et al. [2] developed an interactive system to create 3D building models given a single image with manual assignment of depth using painting metaphor. Sketch based approaches for images to generate facades was presented by Van de Hengel in [3], where the mass modeling was not possible because of high manual intervention involved. There are also vision based methods for architectural modeling from images. Some examples are [4][5]. However, they lack representation of styles in architectural models.

Symmetry and repetitive pattern analysis was used in structure aware facade editing (SAFE) [6]. This framework is based on symmetry and repetitive pattern analysis that provide easy grouping of the facade elements independent of the input decomposition facilitating interactive editing. Symmetry detection as a solution for the hierarchical and layered analysis of irregular facades was proposed recently by Zhang et al. [7] and Mitra et al. [8] presented results on structured symmetry in 3D geometry. Other works on structural symmetry detection extract repeated patterns without organization [9][10]. Martinet recently proposed a structural hierarchy to discover congruent components in a scene in [11]. Another work byWang et al [12] analyzes in a bottom-up manner the symmetry grouping and part assembly guided by heuristic rules. All these approaches focus on optimal editing facility. However, they often generate multiple plausible editable outputs and our work focuses on exploring this variation for 3D shape modeling using interactive editing.

Procedural modeling has been successfully applied to plants [13], cities [14], building [15], and facades [16]. Wonka et al. use split grammars to synthesize building facades with varying styles in [17]. Later, CGA was proposed by Muller et al. [16] especially with the aim of creation of cities. Mass modeling for city construction based on rules derived from a specific design and iteratively generating details by non-terminal rewriting. The CGA can be used for geometric transformations including extruding, splitting, etc. Another work in this area integrates auto-correlation based facade analysis of rectified images with shape grammars [18]. However, the main drawback of this type of modeling approach is the level of expertise in designing rules to describe how the geometric primitives in a model is to be placed in the scene. Therefore understanding the rules and grammars is still necessary to generate and edit models to a certain extent.

Inverse procedural modeling attempts to find a procedural model that generates given input and there has been a lot of research works towards encoding facade layouts. Most recently proposed work by Fuzhang et al. [19] attempts to generate a meaningful split grammar that explains a given facade layout. There are also other works that proposes a grammar that splits a facade into hierarchical components like floors which further hierarchically splits into windows, doors, balconies, etc. [20][21]. This approach reduces the three-dimensional problem to a one-dimensional problem outputting a sequence of elements. Another contribution to this field is the inverse procedural modeling of Stava et al. [22] that finds an L-system that generates an input vector scene. There are also several other important research works that guide in building reconstruction after deriving a shape grammar from input facade layout. For the 3D building reconstruction, it is necessary to use the generated grammar to create sub-facades that fits into complex footprints of the building.

3. METHODOLOGY

Despite important investment on scene representation through primitives, prior art has approached the problem from a questionable angle. The essential questions that one has to answer might be, *'What are the structural components of a scene?, How are the components organized both structurally and architecturally?'*. Finding solutions for these questions are the main motivation for our work. In our approach, we propose a higher-level scene representation like structure aware shape reconstruction techniques to reconstruct the building with less information. Thus, if we can find a way to recover this information from the 2D image, the reconstruction process can be much simpler.

To start with, we describe the 3D building model representation. Then, we describe the structure representation in detail. Further the three cases for applying our scheme; single image view, additional image view, and editable model are discussed.

3.1. 3D Model representation

This section describes the model representation which is specially tailored to facilitate our reconstruction method. This type of representation is chosen because it is found to express most architectural buildings in terms of a small set of parameters which can be recovered using 2D images. Most importantly, it is easy to express constraints of symmetry and alignment within and in between the model elements. This would favour editing the models to a greater extend. The choice of this model representation is to represent the building as a 3D surface model with as few parameters as possible so that the computer can recognize these constraints and reconstruct the model more efficiently. Therefore, we resort to represent the building as a constrained hierarchical structure of parametric primitives called 3D components.

Reconstructing a building from range scanners or structure from motion by acquiring the depth measurement of each of the hundreds of thousands of pixels in a image is a tedious approach. This approach ignores the constrained structure of the architecture. Instead, the building is considered as a cloud of points with hundreds of thousands of individual measurements. In our approach, we use a higher-level representation to reconstruct the building with less information. Thus, if we can find a way to recover this information from the 2D image, the reconstruction process can be much simpler.

The philosophy of this research is that the building is not a collection of point clouds or a set of polygons, but rather it has a very regular structure composed of 3D components. Using higher-level representation, we can model the building in terms of its architectural dimensions rather than co-ordinates of polygonal vertices.

We represent the building as a tree structure 'S' . The nodes of the tree is the atomic shapes. The root of the tree is a specific atomic shape called axiom, which usually represents the footprint of the building. So we can represent axiom as

$$axiom \rightarrow S \tag{1}$$

i.e. we consider the building as a 3D shape of a tree structure 'S' having 3D components like floors; floors divided into door, windows, ornaments and so on as shown in Figure 1. The structure consists of a hierarchy of substructures.

Each sub-structure has a set of 3D components which has definite shape and size.

In the task of generating a structure S of the building, the input is a set of images.

$$I = \{I_1, I_2 ... I_M\} \tag{2}$$

where each single view image I_i is defined as a set of pixels,

$$I_i = \{i; i = 1, ...I_w.I_h\} \tag{3}$$

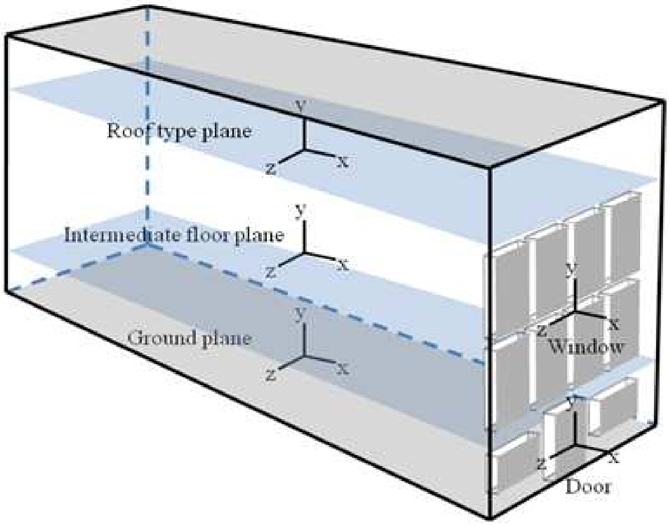


Figure 1: Geometric model of a building, modeled as parametric components.

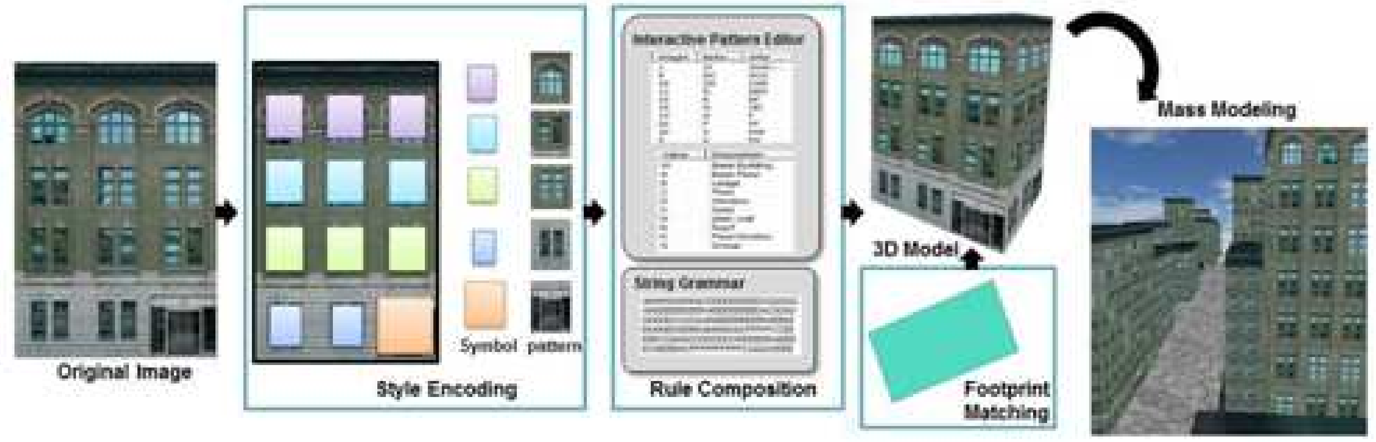


Figure 2: Overall scheme of our proposed modeling and editing.

We assume the image is rectified i.e. the window borders are parallel to image borders and I_w, I_h are image width and height respectively. Next, we explain the three cases for applying our scheme; single image view, additional image view, and editable model.

Case 1: Single image view

Given a single image view, we construct an axis-aligned rectangular box abstraction. Each rectangle encloses an atomic shape of the building. For example, window component, door component, floor component and so on. Computing the abstraction requires detection of atomic shapes and their repetitions. Atomic shapes are spatially linked to each other in the tree structure *S*. The structure *S* is split as floors and roof such that the building is decomposed into floors vertically and floors are horizontally decomposed into components like windows and doors. Each 2D component is converted to its corresponding 3D component by adding a depth parameter *d*, which is the extruded volume of the 2D component. The component is then projected on to the image *I*. Our reconstruction algorithm works by minimizing the error function E_{rri} that is defined by the equation (4).

$$E_{rri} = (F(P_i(S)) - F(I_i))^2 \tag{4}$$

The error function is defined as the matching error between the features projected in the structure S and the features extracted from the image *I*. We define a term P_i which is the projection of the structure S on to the image *I*. Using some features of the structure like component outliers, position etc., we define a term $F(P_i(\mathrm{S}))$which is the feature corresponding to the projection of the structure on to the image *I*. We resort to represent the features as the components extracted as a result of the building structure decomposition. From the decomposition, we can get the component bounding box and material information. We nullify the projection term as we manually project the 2D components to the 3D components. Thus, if we can compute features F in the image view and projected structure, the equation (4) can be readily evaluated. If error is zero, then it is the best match.

Case 2: Additional image view

In this case, additional images are used to refine the reconstruction process. We formulate an optimization to solve for the best matching 3D components in the structure S to the 2D components in the given image view I as a summation over the error function. Therefore, the reconstruction algorithm works by minimizing the error function that is defined by the equation (5). If error is zero, then it is the best match.

$$E_{rr_i} = \sum_i (F(P_i(S)) - F(I_i))^2 \tag{5}$$

Case 3: Editable model

In order to edit an existing reconstructed 3D model, we propose to use rule based grammar formulation. We define an editable model as a transformed structure which varies from the original structure S by some constant value of translation, rotation or scaling. Therefore, for generating an editable model $T(S)$, we define a cost function V_s as in the equation (6):

$$V_s = G(T(S) - S) \tag{6}$$

where G is the rule based grammar that is applied on the structure *S*. Grammar G is a sequence of rules and we define split rule and repeat rule to handle the sub-structure decomposition and repetition respectively. If V_s=0 or V_s<ϵ , then the cost function V_s can be minimized.

3.2. 3D Reconstruction procedure

Real-world buildings have their unique structures with complex footprints with architectural components repetitive along the horizontal and vertical directions. Inspired by this observation we use a user-assisted approach to encode the facade layout as a repetitive component tree.

Our pipeline starts from a single structure analysis to extract the shape information given as input a single non-calibrated 2D image. We then, cluster similar components together in a bottom-up manner in order to label the facade into semantically meaningful architectural components. This step gives sufficient information to generate rules for individual component configurations. Rules for the structural topology are derived based on the hierarchically arranged component tree and geospatial information.

The key observation of our work is that component identification, grouping and solving for individual component fitting in a hierarchical and layered structural representation, leads to an enhanced accuracy and editing capability. Although a fully automated solution for facade encoding exists, to the best of our knowledge, no framework allows for generation of a complete building from the facade and the building footprint. In our work, we propose an interactive user-assisted solution. Beyond the 3D building reconstruction, our framework reveals the possibilities for a layered 3D facade generation, in which signboards and other functional metadata can be placed at the exact positions bringing a real life-like appearance to buildings. In addition, real footprint of the building is given as prior information in order to align the reconstructed facade geometry so that it reflects the real world. The height information is obtained either from real measurement or estimated measurement calculated from the ratio of pixels occupied by the horizontal footprint segment and the pixels occupied by the vertical height of the building image.

In this research work, we focus on the problem of rapid generation of a city based on constrained procedural modeling that adapts to the real footprint data. In this work, we propose a semi-automatic framework to rapidly reconstruct a 3D building model that exactly matches the building photograph taken from a single view using limited prior information. Facade components are extracted from the facade layout and organized as a repetitive shape tree. A meaningful grammar representation is automatically extracted from the hierarchical facade subdivision. We extend the previous approaches of procedural building models to a constraint-based framework for the recovery of the hidden parts of the building. This is an independently done work from [23] although there are some significant similarities in both the approaches. Our work focuses more on updating a real-world building using single facade layout and building footprint. The limitation of the grammar in [23] is that it works only for facade layouts that can be split by a single line. We have extended the grammar formulation to a more flexible and compact form that is suitable to a variety of real-world buildings. We then provide an interactive editing process for updating of the structural topology given a different view of the building. We demonstrate our framework on several real-world buildings with challenging footprints and we show that the procedural representation can generate similar buildings to the original that are used to populate a virtual city. Figure 2 shows the overall workflow.

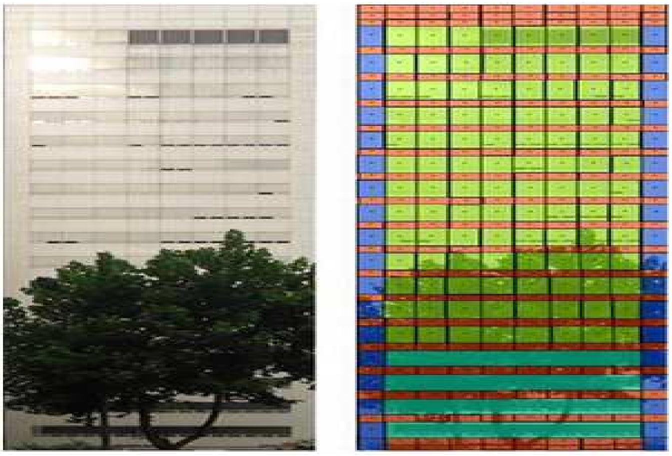


Figure 3: Facade layout given as a labeled segmented image.

Style encoding

The goal of this stage is to capture the structural information in a facade and hierarchically subdivide it. The input is a single image and the output is hierarchically subdivided component groups forming a shape tree structure. This shape tree also contains information about the location of each component in the facade and their inter dependence. Figure 3 shows an example of a subdivided facade computed by our algorithm; Sample input image (left). Facade layout given as a labeled segmented image (right). Although this example is fairly symmetric, it is already challenging due to the different lighting, shading and different window appearance. The algorithm starts with a semi-automatic way of component detection and component grouping. Next, a repetitive shape tree is derived from the hierarchical grouping of the components. This process is useful for the deriving rules from the facade.

Facade layout representation

In this stage, we generate input facade layouts with focus on the structural analysis. Detection of component and their repetitions were experimented earlier in the works of Teboul et al. [24] and Shen et al. [25] but the results are not satisfactory on complex and irregular facades. Therefore, we developed a semi-automatic interactive tool for structural analysis of facade.

Components are defined inside a rectangular domain represented by non-overlapping rectangular regions. Each rectangle encloses a component of the facade which is indivisible throughout the analysis (e.g., window, door, balcony or wall). A rectangular region R is defined by its position of the x and y co-ordinate in the lower left corner (x_i, y_i) and upper right corner (w_i, h_i) of the rectangle. For finding the repetition, the user draws a rectangle over any one of the component. The system automatically finds the repetition of that particular component in the facade using normalized pixel-wise square difference measurement. In order to compensate the error in similarity measurement caused by appearance variation among repeated elements, we also allow for manual verification and adjustment of the matching results. The rectangles that are deemed to be repetitions are labeled by the same color. A labeling function $L(a,b)$, describes the material at position (x_i+a, y_i+b) as an integer label which in turn is visualized by the same color.

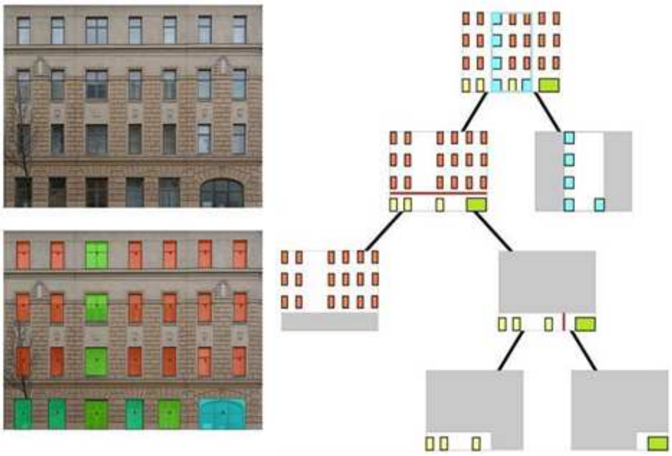


Figure 4: Repetitive shape tree derived from the given image.

Repetitive shape tree

In this stage, we organize the labeled components hierarchically into a shape tree. In order to structure the architectural components hierarchically we first iteratively split the facade horizontally and vertically in a top-down manner. This hierarchical structuring technique using layered facade analysis by symmetry maximization was adapted from the work of [26] . We re-implemented this technique for component recognition and further in rule derivation. In every splitting step, the components are sorted by their lengths along the splitting direction in the decreasing order. Next step is to cluster the components into groups. This process organizes the components into a structured shape tree represented as $G(V,E)$ where V is the component groups and E is the relation in position between the component groups. Figure 4 shows the derived shape tree from the given facade.

In order to minimize the energy of the repetitive shape tree $G(V,E)$, we define the energy function $E(L)$ as in equation (7):

$$E(L) = min\left\{ \sum_{i\epsilon V} E_s(l_i) + \sum_{(i,j)\epsilon E} E_c(l_i, l_j) \right\} \tag{7}$$

where l_i is the label of the i-th component and E_s and E_c are the similarity energy and coherence energy respectively.

E_s describes the shape similarity to its labeled type l. For example, the ratio of width to height of window is relatively smaller than the door, even though both are openings on the wall. The repeated frequency of windows is high while the door is low.

Therefore, the shape similarity to a certain component type is evaluated by the consistency between the feature values of the component and the component type. E_c describes the location coherence of components with labels l_i and l_j representing their inter dependencies. For example, a wall indicates that there is a much higher probability of having a window rather than a door above it. Therefore, the location coherence is evaluated based on the relative locations of adjacent pairs of components in the vertical and horizontal directions. We use dynamic programming to solve the energy on the shape tree, $E(L)$. However, in cases of vertex merging, the exact minimization is solved using belief propagation [27], [28]. During energy minimization, the labels of known vertices are used to infer the labels of unknown vertices and its corresponding subdivisions.

Rule Composition

At this stage of the pipeline, the structural information of the facade is encoded as a repetitive shape tree. As the next step, we encode the computed shape tree as shape grammar rules [29]. This is based on the observation that a building consists of a regular pattern that could be exploited to automatically construct a representative grammar. The rule set for the facade is encoded as string grammar [30]. Our algorithm exploits this global outlook to derive rules for reconstructing the building. The rules for the facade structure including floors and tiles are encoded as combination of split and repeat operations. For instance, the building can be subdivided vertically as several floors; each floor is further subdivided into various tiles representing walls, windows, doors for example. Within a single building, there can be different types of floors, a variety of window styles and trims and several types of wall materials such as brick, stone, and so forth.

A facade S can be represented by a production rule, containing symbols for each individual component. First, the facade is divided into floors. Each floor can be further divided into tiles.

$FacadeS \rightarrow (GroundFloor)(IntermediateFloors)(Roof)$
$IntermediateFloors \rightarrow (F_1F_2..F_M)$
$FloorF_i \rightarrow (T_1T_2..T_P)$

Two tiles in a floor are considered similar if the labels of their respective terminals are same and depicted with the same color.

Grammar parsing

Given the hierarchically subdivided facade layout as input, the system automatically finds the repetitions in building facade and creates a set of production rules. Production rules consist of lexical elements known as non-terminals and terminals. Non-terminal symbols are those that can be replaced by either non-terminals or terminals. Terminals cannot be subdivided further. For parsing, all buildings are assumed to be organized from bottom to top. The first floor is considered as the ground floor. The roof is the top floor. The intermediate floors are considered as repeatable floors. This organization defines a hierarchy of production rules where the grammar terminals can be re-

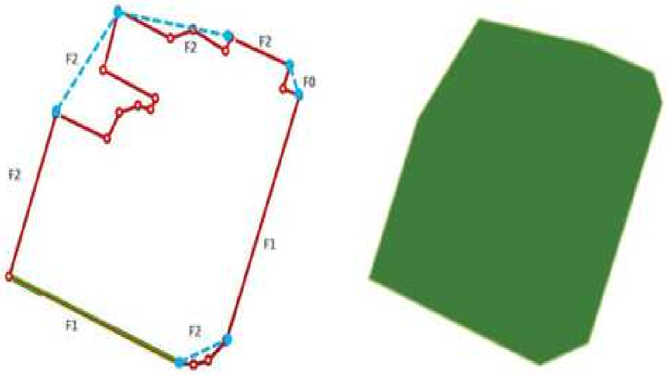


Figure 5: Conceptual diagram illustrating the elastic band footprint alignment.

placed either by materials of components or its 3D models in the rendering phase.

Grammar parsing automatically gains the needed instantiations of these rules to describe the arrangement of each component of a building. This also helps later in interactive editing of the components by increasing or decreasing the number of floors by adapting to the production rules. Thus, given as input a hierarchically subdivided facade layout, our algorithm can automatically infer production rules to reconstruct the building as real as possible and also yield new and similar-in-style buildings.

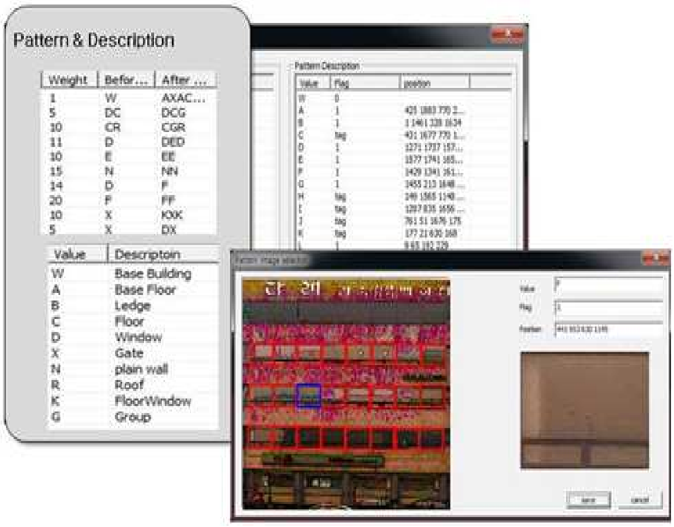
Footprint alignment rule

Footprint is also given as an input and it represents correct alignment of each generated facade. Our method chooses the best production rule to apply a footprint production rule to a facade. Each building consists of several facades. Each facade has several floor productions $F_i = (F_1, F_2 \ldots F_M)$ where F_i is connected horizontally to $F_{(i+1)} \bmod M$. The base floor is aligned according to the footprint rule. Each footprint is composed of several segments with unequal lengths and complex orientations. Each facade has two adjacencies and hence many categories of facade layouts. The challenge here is to find the best candidate footprint production rule that will match with the facade layout. As a solution, we propose the elastic band technique inspired from the idea of using Nudged elastic band (NEB) [31] for finding the minimum energy paths for transitions in quantum dynamics. This method works by optimizing the number of intermediate saddle points by several iterations along the path that connects two footprint segments. Each iteration finds the lowest energy possible while connecting to the neighboring segments. This constrained optimization is done by adding spring forces along the band between segments and projecting out the segment that is appropriate in terms of corner orientation and width of the facade. Figure 5 shows the conceptual diagram illustrating the elastic band technique. The dotted lines shows the merged intermediate footprint segments thereby minimizing the paths between the neighboring segments without affecting the global appearance of the building.

In order to increase the flexibility and expressivity of our grammar we provide semantic information about the building as part of our grammar which we name as attribute grammar. This semantic information can be the total number of floors in the building, total number of visible facades of the building in the image view, the number of repeating floors and so on. At the present stage, we have provided only limited semantic information, which can be extended in future to include more flexibility. This type of semantic information is helpful in reconstruction when only part of the building image is given. Thus, our framework provides the flexibility to create a building structure from limited information combining both structural information and semantic information about the building.

Interactive pattern matching

At the previous stage, we have matched each of the component clusters with a symbol specified in the shape grammar. This component cluster is a set of rectangular regions clustered into groups of similar components. At this stage we need to match the components to a pattern image. This is useful for generating high quality geometric information that can provide some semantic interpretation. We have developed an interactive pattern editor for matching a pattern image to the component cluster. For example, a window pattern is assigned to a symbol representing the window. For each component cluster, there are three attributes: value, flag and default position. The value is the symbol representing the individual component. The flag field is 0 if there is no pattern, 1 if it contains a pattern and tag if the component is a signboard. The tag flag is important as it provides for an extra layer over the facade layout. The pattern editor tool is shown in the Figure 6. This can be extended in future, to include 3D window model as replacement to the window pattern image. The facade layout contains information about the



Pattern & Description

Weight	Befor...	After ...
1	W	AXAC...
5	DC	DCG
10	CR	CGR
11	D	DED
10	E	EE
15	N	NN
14	D	F
20	F	FF
10	X	KXK
5	X	DX

Value	Descriptoin
W	Base Building
A	Base Floor
B	Ledge
C	Floor
D	Window
X	Gate
N	plain wall
R	Roof
K	FloorWindow
G	Group

Value	Flag	position
W	0	
A	1	425 1883 770 1...
B	1	1 1461 328 1634
C	tag	431 1677 770 1...
D	1	1271 1737 157...
E	1	1577 1741 165...
F	1	1429 1341 161...
G	1	1455 213 1648 ...
H	tag	149 1365 1148 ...
I	tag	1297 835 1656 ...
J	tag	761 51 1676 175
K	tag	177 21 630 168
L	1	9 65 182 229

Figure 6: Interactive pattern editor tool.

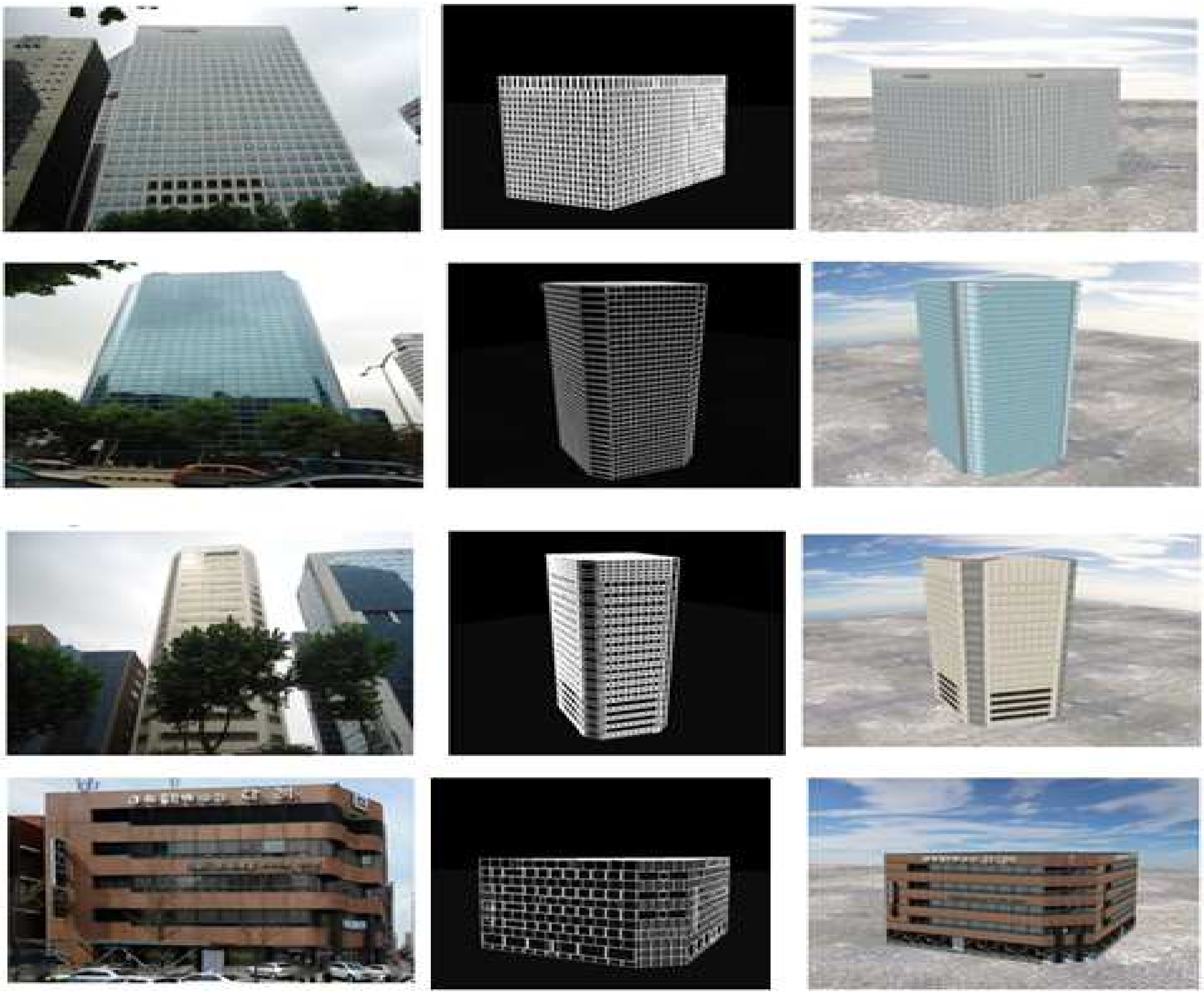


Figure 7: 3D reconstructed results of real city buildings.

locations of components and its pattern image. Thus by default, each of the components in the cluster has a pattern image determined by its location in the facade layout. Using the pattern editor, the user can select a template image pattern, T to find the best fitting pattern image in the component cluster.

The system then automatically assigns the best matched pattern to the entire component cluster. We compute the best matching pattern image I_p per cluster as expressed in equation (8):

$$I_p = argmax \sum_{T \epsilon C} S(I(R_i), I(T)) \tag{8}$$

where S is the similarity measure on image intensities at corresponding positions inside the rectangular regions in the component cluster denoted by R_i. Hence, the best-matching image pattern is selected. The image pattern is then back projected to the entire component cluster. Additionally, material effects like shader information for reflecting glass can also be given to the patterns. This increases the realistic appearance of reconstructed 3D buildings.

4. Results and Discussion

We have chosen single view images of four candidate buildings located on the real street at a modern city to show the flexibility of our approach. The captured buildings are tall city buildings and have different footprint complexity. Case 1 is a simple cube style building. Case 2 and Case 3 are buildings with small sub-sides on each side. Case 4 is a building with complex foot prints where facades are not planar or convex and has hidden parts of the facade which are not visible in the taken image. The specifications of the buildings are shown in Table 1.

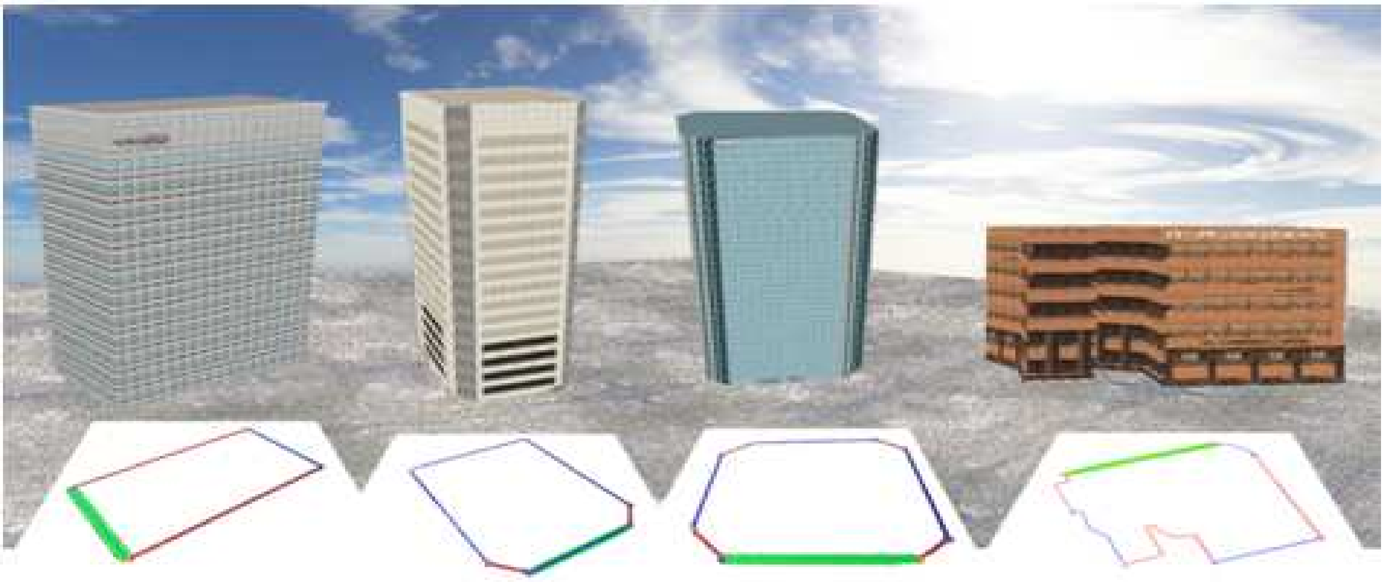


Figure 8: Reconstructed 3D models of real buildings and respective footprint alignment.

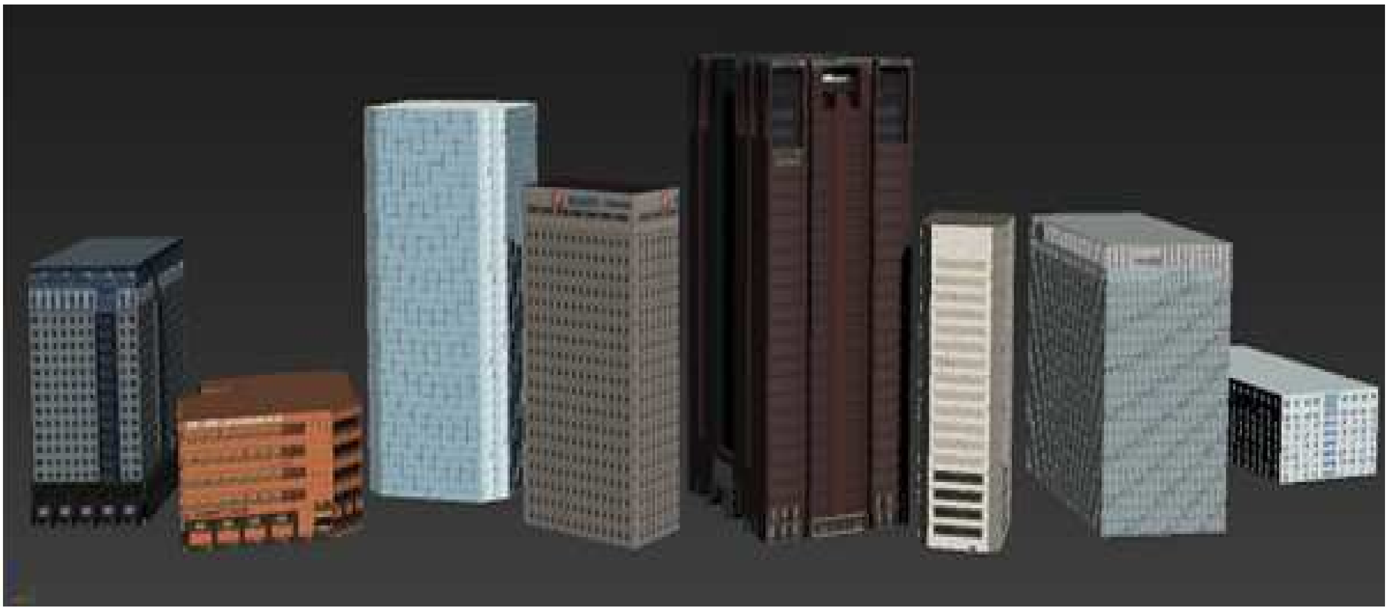


Figure 9: Final reconstructed buildings exported and viewed in 3DS MAX.

The facade view is tilted as it is difficult to capture tall buildings in full. Therefore images were pre-processed to rectify them to get a full frontal view of the building. Rectification can be done by cropping the captured images. In the experiments, cropping is done using general interactive cropping tools. It is also possible to conceive an automatic method by selecting boundary edges from the recognized components in the image. In the style encoding stage of our proposed method, we used openCV and Qt framework to detect the repetitions of components in the image. Next, the string grammar was automatically generated by our system. Interactive pattern editor was developed to select and match the components to the facade layout in the rendering stage. 3D model was reconstructed and rendered using OpenGL framework. The results of the 3D reconstruction are shown in Figure 7; Original Image, 3D wireframe rendering, 3D photorealistic rendering of the building in a sequence from left to right of Case I, Case II, Case III, Case IV candidate buildings respectively. The aligned 3D reconstructed buildings to their respective footprint shape is shown in Figure 8. Figure 9 depicts the final reconstructed buildings exported and viewed in 3DS MAX which can further be utilized for any architectural design purpose or for 3D games.

An important aspect of our application is mass modeling using same family of buildings and updating a virtual city with different style buildings. Using our framework, we can generate variations of the same family of buildings and populated a virtual street, which can be extended to a virtual city modeling. To obtain the variations of buildings, we generated grammar variations using interactive editing of automatically extracted grammars. The second application of our approach is that we can detect the exact dimensions, positions, spacing and ratios of architectural components automatically. This method can be utilized by commercial urban model providers to create 3D building volumes. The third general application of our approach is 3D building modeling for games, entertainment, personal web and user content creation (UCC). Figure 11,12 show some of the application domains.

For a 3D building to appear realistic, the height, width and depth should also provide real world like feel. The height information is either obtained from real measurement or estimated by finding ratio of pixels occupied by the horizontal footprint segment and the pixels occupied by the vertical height of the building image. We select one of the footprint segments as the front side, by making the best guess to fit into the captured image. We then adjust the width of the front facade of the 3D model to the width of the front side of the footprint segment. Thus, a ratio of the width/height of the facade image resized is maintained throughout all facades of the 3D model that is rendered. The reconstructed 3D building height is compared with the real height data. The results in Table 1 show that there is considerably small height difference in the measured height data.

Our approach has several limitations for improvement as future work. The scope of our facade sub-division step is limited to semi-automatically identifying the components of a facade using vision algorithms and manual refinement of the wrong results. On a technical level, our facade subdivision scheme can be enhanced by utilizing the recursive sub-division scheme proposed in [33]. For example, applying clustering techniques to the over-segmented patches in the facade image can produce reliable segmentation results which is still a major challenge in the procedural modeling pipeline.

Another limitation is that the automatic depth reconstruction technique may fail when trying to model a 3D structure procedurally with components. The other problems are concerned with the image noise and irregular protruding structures in the building facades in addition to the general components that is a barrier for fully automating the pipeline. Also, the ground floors of many city buildings are often difficult to detect and model. We have observed that, our approach is not a good solution for modeling less repetitive architectural structures. Hence, our approach is more feasible for multi-storied buildings with repetitive facade components.

We experimented modeling Taj Mahal with our proposed methodology. The middle facade portion of the building can be reconstructed using our approach. But the round shaped dome is not yet being able to be reconstructed by our approach. Figure 10 shows the ill reconstructed Taj Mal model which clearly highlights the limitation of our current approach.

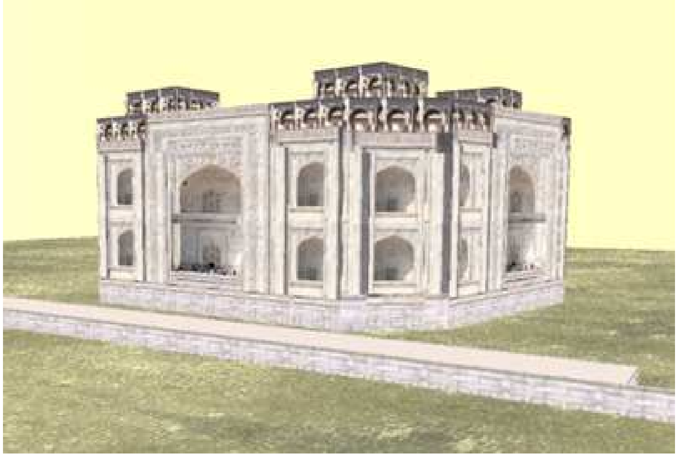


Figure 10: Ill reconstructed Taj Mahal model using our approach.

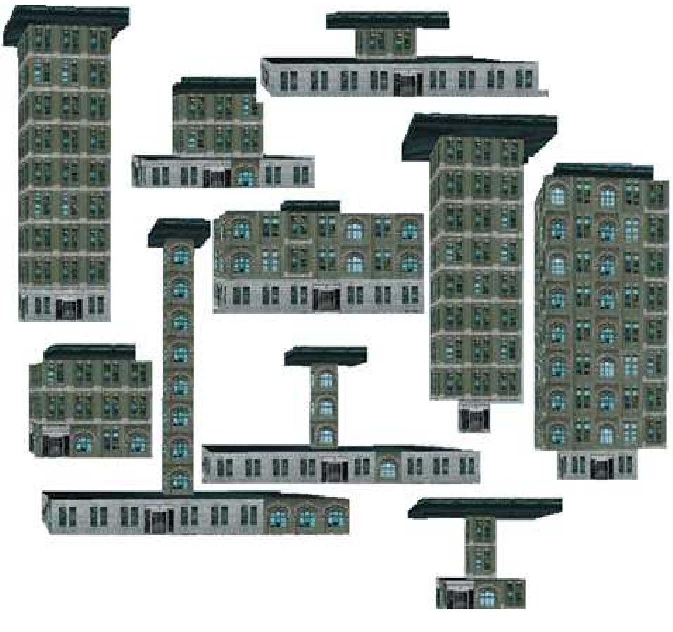


Figure 11: Numerous potential solutions from a single facade layout.

4.1. *Occlusion recovery*

We use production rules and multiple instances of a terminal type to replace parts of the building that are not visible in the original image. We can distinguish two types of occlusions. Occlusions that are caused by objects in front of the target facade like trees, pedestrians, vehicles, telegraph poles and trash cans. Another type of occlusion is the hidden views of the building like side views and back view including sub-facade hided in the view. In order to fully reconstruct a building, we use inverse projection

Table 1: Comparing the real height of buildings with measured height of 3D model.

Building	No.of floors	Real height(m)	Measured height of 3D model	Error (%)
Case I	24	108.45	108.643	0.177
Case II	21	84.15	77.165	8.30
Case III	18	81.49	78.420	3.761
Case IV	5	26.1	21.507	17.59

technique. Base idea is to render occluded facade with terminals from a selected subset of the image that best matches the occluded portion. For a occluded terminal T_i, we define a fitness function $F(T_i,\ P(V_i,\ T_j))$ to determine the best match between the terminal and its possible replacements from the view V_i using non-occluded terminal T_j of the same component group where P is inverse projection function of facade terminals to 3D space. There are some assumptions to be considered while determining the best fit. The sum of terminal sizes of T_i and T_j should almost be the same with the length of the footprint. As an assumption, T_j is generated by the extension of the rules from the facade that shares the similar directions. Thus, an optimization procedure is to generate rules from the facade rule sets which minimizes the following function among T_js generated from the facade rules which shared similar directions, where $L_{footprint}$ is the length of footprint boundary and $W(T_i)$ is the width of a terminal. The fitness function is expressed in equation (9).

$$F = min\left\{\sum W(T_i), W(T_j)) - L_{footprint})^2\right\} \tag{9}$$

Figure 12: Generating virtual street with different building variations and footprints.

The 3D model is reconstructed based on a single view image. We also experimented on how to update the 3D model given the input facade layout of the view taken from a different view of the building. We first capture images from two adjacent views of a building. We then, update the hidden side according to the facade layout in the second image by providing generated facade rules from the second image as a candidate sets for hidden terminals as in the equation (6). The results in Figure 13, (Top left) the image taken for a building, (Top right) reconstructed 3D model from the image, (Bottom left) the new image of the same building taken from the different view, (Bottom right) the updated 3D building.

4.2. Comparative evaluation

We have compared our approach with the most recent procedural modeling approach [19] which was presented in SIGGRAPH 2014. Our research work is an independently done work from [23] although there are some significant similarities in both the approaches. Wu et al. generates meaningful split grammar that explains a given facade layout and use the grammar to generate variations for large scale urban modeling. Our work focuses more on updating a real-world building using single facade layout and building footprint. The grammar extraction and parsing are similar in both the approaches. The limitation of the grammar in [23] is that it works only for facade layouts that can be split by a single line. We have extended the grammar formulation to a more flexible and compactible

Figure 13: Updation process of 3D building given a different view.

form that is suitable to a variety of real-world buildings. We use attribute grammar that is made more flexible and expressive by combining both structural information and semantic information about the building structure. E.G., we can specify the number of floors, total number of visible facades of the building in the image view, the number of repeating floors and so on. In our approach, updating reconstructed 3D model using image captured from different view point is possible. This is not considered in the previous approach. Size optimization is done by estimating the height-width ratio of 3D model optimized using image height-footprint segment width ratio. Size optimization is not considered in the previous approach. In both approaches massive modeling is supported by procedural technique. In our approach, automatic footprint alignment of the reconstructed models is possible, which is not considered in previous approach.

In Ref.[32] work, describes its own limitation such that the meaningfulness of their reconstructed results depends on the meaningfulness of the box hierarchy. We experimented with the Case IV building with non-convex footprint using Lin's approach and our proposed approach. The results are shown in Figure 14. We improved from Lin's approach by considering feature alignment via interactive editing in addition to box alignment. The visual appearance of the result clearly shows the improvement in our proposed approach by enhancing the height distribution and alignment schemes. Wu's approach also [23] cannot handle this type of facade layouts as their grammar can only handle layouts that can be split by a single line.

Another limitation as described in [32] is that their method is inherent to the fact that it is developed for retargeting and not generic modeling. Our approach supports both generic modeling given the photograph of a building and also retargeting to create variations of the reconstructed 3D model.

In addition to the irregularity in building structure mentioned in Lin's approach, we have also targeted to solve for complex and non-convex footprints as shown in Case IV reconstruction results.

In [33] work, they consider the sequence of overlapping images captured along streets, and automatically compute the structure from motion to obtain a set of semi-dense 3D points cloud and all camera positions. They, then register the reconstruction with an existing approximate model of the buildings using GPS data if provided or manually if geo-registration information is not available.

The facades of buildings are decomposed by segmenting it to a minimal number of elements. Their approach considers all visible horizontal and vertical lines to construct an irregular partition of the facade plane into rectangles of various sizes. This partition captures the global rectilinear structure of the facades and buildings and also keeps all discontinuities of the facade sub-structures. Still, such a partitioning method is subject to limitations when we consider its application to the real world buildings. Their partition schema is subject to segmentation parameters which usually results in over-segmentation of the image into patches. In order to minimize such an over segmentation, further optimization of the recursive subdivision method is needed and also in [33], they utilize additional information for partition from the pre-generated dense 3D point cloud of buildings.

In our approach, we use limited information like single or limited number of 2D images of a building as input rather than first creating a 3D point cloud of the input data. Creating point cloud from range scanners or structure from motion by acquiring the depth measurement of each of the hundreds of thousands of pixels in a image is itself a tedious approach.

In our facade decomposition method, we structure the architectural components hierarchically by first iteratively splitting the facade horizontally and vertically in a top-down manner. This hierarchical structuring technique using layered facade analysis by symmetry maximization was adapted from the work of [26]. We re-implemented this technique as a semi-automatic and interactive method for component recognition and further in rule derivation. Such a semi-automatic and interactive component identification process is preferred to a fully automated segmentation process due to the following reasons. (1) It is fully training free; (2) it is much easier to make a guess on the hierarchy or the organization of component structure; (3) If the component identified by the system is wrong, then manual refinement is possible which guarantees good reconstruction results. Also when considering the splitting operation in our approach, in every splitting step, the components are sorted by their lengths along the splitting direction in the decreasing order. This eliminates any over-segmentation of the facade image into patches and therefore further optimization is not required. In this way, our algorithm is independent of the recursive subdivision method used for facade decomposition in [33].

4.3. User Study

In order to evaluate the user experience for the proposed system, we selected fifteen subjects. We divided the subjects into two groups of which, one group consists of novice subjects who have experience of 3D modeling (Category 1) and another group of non-experts (Category 2).

Figure 14: Comparison of Lin's approach (left) and our proposed approach (right).

Table 2: Results of the modeling experiment.

	Category 1			Category 2		
	$UserA$	$UserB$	$UserC$	$UserD$	$UserE$	$UserF$
Number of generated buildings	40	33	34	26	21	23
Total time (sec)	3800	3600	3500	3600	3500	3700

We provided each user with the facade layout and images of the facades used in our case study. They were given two tasks. First task was to reconstruct a building from any of the facade layout. Second task was to generate a virtual city from any desired building footprint of the users choice. Every group of the participants has a meeting before starting their experiment. In the meeting we introduced objectives and modeling guidelines of the experiment. The modeling session stops when each user has made a set of closed models that resembles the original 2D image.

Each participant of the group performed their modeling task according to the given 2D image and the facade layout. Table 2 shows the result the experiment. According to Table 2, the participants showed average results of the proposed framework.

Next, we conducted a survey on the participants of the previous experiment (15 subjects) for the sake of satisfaction of the proposed scheme. The questionnaire consists of 11 questions based on the visualization and easiness in using our developed user interface for 3D building generation. The answers consist of 5 possible answers: strongly agree, agree, normal, disagree and strongly disagree. The results of the survey show the proposed scheme is satisfied to various areas of measuring points. Especially, the participants said the proposed scheme is recommendable for city generation and urban designing. The participants also answered that if the proposed system can be applied to their research applications in virtual reality and augmented reality fields, then it is feasible to utilize the proposed mechanism in the partial parts of entire modeling process.

We also calculated Cronbachs alpha value from the results of survey in order for proving reliability. The Cronbachs alpha value is $0.86 > 0.7$, we thus argue the results of survey is statistically significant. We also interviewed the navigation company people who are map service experts. They have reported that our approach is quite reliable and useful. In addition, they have mentioned that the visual quality of our approach is still marginal. As future work, we will consider their suggestions and improve the visual quality using revised rendering methods.

5. CONCLUSION

In this paper, we have presented a modeling framework to generate and update real world 3D building given as input a single facade layout and real footprint data. Using several examples, we demonstrated that 3D buildings can be modeled quickly and rendered with realism or style compared to the original structures. We also demonstrated that the extracted procedural rules can be easily adapted during interactive editing to create models with variations.

As future work, we would like to extend our approach to non-rectangular facades having dome or curved shaped architectures like ancient Mughal monuments. Also, we target to model other irregular pyramid shaped skyscrapers like the Brazilian Tower. Although a complete accurate 3D model is not possible at this stage using our framework, a coarse structure can be estimated which can further be edited using extensive manual methods. We also suggest that using sketch based editing methods i.e user can draw some side view sketches to fill in the missing parts of the coarsely reconstructed building models. Further, we are exploring the possibility for combining our system with an automated city modeling system, for generating virtual urban spaces in the style of an existing city. We are investigating methods for extending our work to incorporate more styles of buildings with different roof patterns and structural complexity. The existing editing process could be enhanced in future by selecting a component cluster and enhancing the shape interactively. Finally, in order to provide practical feasibility of our approach in the real world scenario, we would like to investigate the possibility of integrating our 3D building modeling approach with the existing navigation systems like Daum Road View.

References

[1] Debevec P., Taylor C., Malik J.: Modeling and rendering architecture from photographs: a hybrid geometry and image-based approach. In *Proc. SIGGRAPH '96* (1996), pp. 11–20. doi:10.1145/237170.237191.

[2] Oh B. M., Chen M., Dorsey J., Durand F.: Image-based modeling and photo editing. *ACM TOG* (2001), 433–442. doi:10.1145/383259.383310.

[3] Hengel A. V. D., Dick A., Thormahlen T., Ward B., Torr P. H. S.: Videotrace: rapid interactive scene modeling from video. *ACM TOG 26*, 3 (July 2007). doi:10.1145/1276377.1276485.

[4] Snavely N., Seitz S. M., Szeliski R.: Photo tourism: Exploring photo collections in 3d. *ACM TOG 25*, 3 (Aug. 2006), C835–C846. doi:10.1145/1141911.1141964.

[5] Goesele M., Snavely N., Curless B., Seitz S. M., Hoppe H.: Multi-view stereo for community photo collections. In *Proc. IEEE ICCV'07* (2007), pp. 1–8. doi:10.1109/ICCV.2007.4408933.

[6] Minh D., Duygu C., Boris N., Mark P.: Safe:structure-aware facade editing. *Computer Graphics Forum*, 2 (2014), 83–93. doi:10.1111/cgf.12313.

[7] Zhang H., Xu K., Jiang W., Lin J., Cohen-Or D., Chen B.: Layered analysis of irregular facades via symmetry maximization. *ACM TOG 32*, 4 (July 2013), C104:1–C104:13. doi:10.1145/2461912.2461923.

[8] Mitra N. J., Pauly M., Wand M., Ceylan D.: Symmetry in 3d geometry: Extraction and applications. In *Proc. EUROGRAPHICS State-of-the-art Report'12* (2012). doi:10.1111/cgf.12010.

[9] Pauly M., Mitra N. J., Wallner J., Pottmann H., Guibas L.: Discovering structural regularity in 3d geometry. *ACM TOG 27*, 3 (Aug. 2008), 1–11. doi:10.1145/1360612.1360642.

[10] Wu H., Wang Y., Feng K. C., Wong T. T., Lee T., Heng P. A.: Resizing by symmetry-summarization. *ACM TOG 29*, 6 (Dec. 2010), C159:1–C159:10. doi:10.1145/1882261.1866185.

[11] Martinet A.: *Structuring 3D Geometry based on Symmetry and Instancing Information.* PhD thesis, INP Grenoble, 2007.

[12] Wang Y., Xu K., Li J., Zhang H., Shamir A., Liu L., Cheng Z., Xiong Y.: Symmetry hierarchy of man-made objects. *Computer Graphics Forum 30*, 2 (2011), C287–C296. doi:10.1111/j.1467-8659.2011.01885.x.

[13] Prusinkiewicz P., Lindenmayer A.: *The algorithmic beauty of plants.* Springer Verlag, 1990.

[14] Parish Y. I. H., Muller P.: Procedural modeling of cities. In *Proc. SIGGRAPH '01* (2001), pp. 301–308. doi:10.1145/383259.383292.

[15] Muller P., Wonka P., Haegler S., Ulmer A., Gool L. V.: Procedural modeling of buildings. *ACM TOG 25*, 3 (July 2006), 614–623. doi:10.1145/1141911.1141931.

[16] Muller P., Zeng G., Wonka P., Gool L. V.: Image-based procedural modeling of facades. *ACM TOG 26*, 3 (July 2007). doi:10.1145/1276377.1276484.

[17] Wonka P., Wimmer M., Sillion F., Ribarsky W.: Instant architecture. *ACM TOG 22*, 3 (July 2003), C669–C677. doi:10.1145/882262.882324.

[18] Teboul O., Kokkinos I., Simon L., Sourakis P. K., Paragios N.: Shape grammar parsing via reinforcement learning. In *Proc. CVPR'11* (2011), pp. 2273–2280. doi:10.1109/CVPR.2011.5995319.

[19] Wu F., Yan D. M., Dong W., Zhang X., Wonka P.: Inverse procedural modeling of facade layouts. *ACM TOG 34*, 4 (July 2014), C121:1–C121:10. doi:10.1145/2601097.2601162.

[29] Bekins D. R., Aliaga D. G.: Build-by-number: Rearranging the real world to visualize novel architectural spaces. In *Proc. IEEE VIS '05* (2005), vol. 19, pp. 143–150. doi:10.1109/VIS.2005.13.

[21] Aliaga D. G., Rosen P. A., Bekins D. R.: Style grammars for interactive visualization of architecture. *IEEE Trans. on Vis. and Comp. Graph 13*, 4 (July 2007), C786–C797. doi:10.1109/TVCG.2007.1024.

[22] Stava O., Benes B., Mech R., Aliaga D. G., Kristof P.: Inverse procedural modeling by automatic generation of l-systems. *Computer Graphics Forum*, 2 (2010), C665–C674.

[23] Wu F., Yan D. M., Dong W., Zhang X., Wonka P. Inverse procedural modeling of facade layouts. *ACM TOG 34*, 2014. C121:1C121:10.

[24] Teboul O., Kokkinos I., Simon L., Sourakis P. K., Paragios N. Shape grammar parsing via reinforcement learning. *Proc. CVPR11*, 2011. 22732280.

[25] Shen C.H., Huang S.S., Fu H., Hu S.M. Adaptive partitioning of urban facades. *Proc. of ACM SIGGRAPH Asia*, 2011, Issue 6, Vol. 30. 184.

[26] Zhang H., Xu K., Jiang W., Lin J., Cohen- Or D., Chen B. Layered analysis of irregular facades via symmetry maximization. *ACM Trans on Graph*, 2013, Issue 4, Vol. 32. C104:1-C104:13

[27] Yedidia J., Freeman W., Weiss Y. Understanding Belief Propagation and Its Generalizations. *Morgan Kaufmann Publishers*, 2003

[28] Bishop, C. Pattern Recognition and Machine Learning. *Springer Verlag*, 2006

[29] Bekins D. R., Aliaga D. G. Build-by-number: Rearranging the real world to visualize novel architectural spaces. *Proc. IEEE VIS 05*, 2005. 143150.

[30] Muller P., Wonka P., Haegler S., Ulmer A., Gool L. V. Procedural modeling of buildings. *ACM TOG 25*, 2006. 614623.

[31] Jonsson H., Mills G., Jacobsen K. W. Classical and Quantum Dynamics in Condensed Phase Simulations. *ACM TOG 25*,1998.

[32] Lin, J., Cohen-Or, D., Zhang, H., Liang, C., Sharf, A., Deussen, O., and Chen, B. Structure-preserving retargeting of irregular 3D architecture. *ACM Trans. on Graph*, 2011, Issue 6, Vol. 30. 183:1183:10.

[33] Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E., Quan, L. Image based facade modeling. *ACM Trans. on Graph*, 2008, Issue 5, Vol. 27. Article 161.