EUROGRAPHICS 2015 / O. Sorkine-Hornung and M. Wimmer
(Guest Editors)

Volume 34 (2015), Number 2

Template Assembly for Detailed Urban Reconstruction

Liangliang Nan, Caigui Jiang, Bernard Ghanem, and Peter Wonka

KAUST, KSA

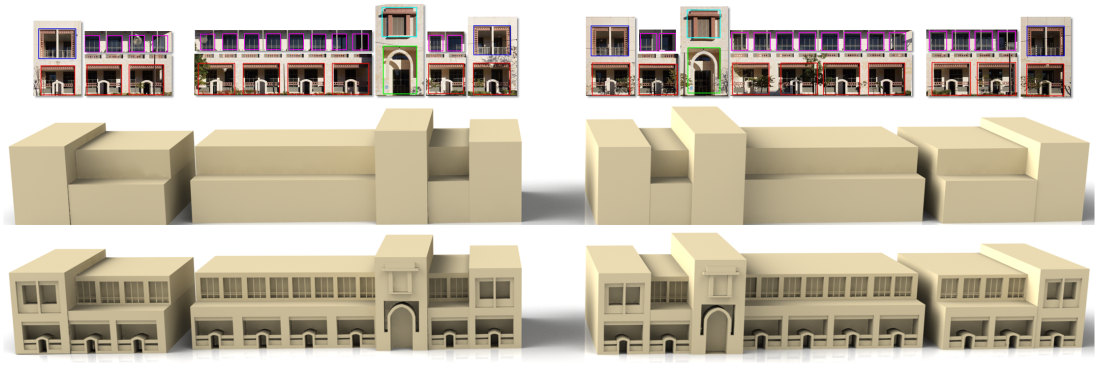


Figure 1: *Two buildings reconstructed with details using our template assembly approach. From top to bottom: the template locations marked in the optimized texture images, the coarse model, and the detailed 3D surface model. Different colors indicate different templates.*

Abstract

We propose a new framework to reconstruct building details by automatically assembling 3D templates on coarse textured building models. In a preprocessing step, we generate an initial coarse model to approximate a point cloud computed using Structure from Motion and Multi View Stereo, and we model a set of 3D templates of facade details. Next, we optimize the initial coarse model to enforce consistency between geometry and appearance (texture images). Then, building details are reconstructed by assembling templates on the textured faces of the coarse model. The 3D templates are automatically chosen and located by our optimization-based template assembly algorithm that balances image matching and structural regularity. In the results, we demonstrate how our framework can enrich the details of coarse models using various data sets.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

Reconstruction of urban scenes from a set of images remains a challenge in both computer graphics and computer vision. As a first step, several algorithms can extract point clouds from a set of images using Structure from Motion (SfM) [HZ00]. While these point clouds can be rendered in an impressive manner, they are actually quite sparse. For example, in a typical point cloud reconstructed by Multi View Stereo (MVS) [FP10], fine details of facade elements (e.g., door decorations, window frames, etc.) are usually contaminated by the noise in the point clouds.

The density of the point clouds is still reasonable for generating coarse building mass models (where building facades are approximated by large textured polygons), and various techniques have been developed for this purpose [XFZ*09, VAB10, VAB12, ZN12, ZN13, LGZ*13] (Note that some of these techniques were actually developed for LiDAR data). However, generating high-quality detailed

models remains an open problem. Even the automatic generation of coarse models is not robust and typically requires manual editing [XFT*08, ASF*13]. Furthermore, facade details are reconstructed almost completely manually.

The goal of this paper is to bring more facade details into the image-based urban reconstruction pipeline. Our proposed strategy relies on image information to add more details to the models. Thus, we propose two new algorithms to augment the image-based reconstruction pipeline. First, we improve coarse mass models that are computed by fitting shapes to point clouds by making use of image information (e.g., edges in the facade textures). To this effect, we propose a new optimization algorithm that combines both image and geometry-based energy terms. Second, we place geometrically detailed templates onto the facades of the coarse model using image information as the query key. Given a set of different 3D templates, representing a facade by choosing a subset of the templates is challenging due to illumination variations, occlusions, and the complex nature of facade design. We use an optimization approach that leverages template matching and prior knowledge of the structural properties of facades. In Fig. 1 we show an example reconstruction.

Our paper makes the following contributions:

- a framework that can generate more detailed building models from a set of input images than existing methods.
- a novel geometry-appearance consistent optimization algorithm for coarse models, which enhances the consistency between their geometry and the images.
- an optimization-based template assembly algorithm that faithfully and effectively represents facade structures.

2. Related Work

Recent years, tools and methods for analyzing and reconstructing architectural models has been researched extensively. For a more comprehensive survey of related work, please refer to [VAW*09, MWA*13]. In this section, we review related work in the areas of photogrammetry-based reconstruction, primitive-based reconstruction, and repetition detection and facade parsing.

Photogrammetry-based reconstruction. Works in computer vision mainly focus on obtaining dense point clouds [GSC*07, FP10, WYJT10, WACS10, Wu11, CMZP14] or automatic reconstruction of textured models [WZ02, DTC04, PF08] from collections of photos, relying on photogrammetric reconstruction and image-based modeling techniques. Xiao et al. [XFT*08, XFZ*09] exploit SfM for depth enhanced facade modeling. They assume planar rectangular facades where details are essentially 2.5D rectangular elements on top of them. A facade is decomposed into rectilinear patches, and each patch is then augmented with a depth value from SfM. Wu et al. [WACS12] proposed a schematic algorithm to reconstruct dense mesh models from profile curves extracted from sparse point clouds. Sinha et al. [SSS*08] present an interactive image-based modeling system for reconstructing piecewise planar 3D structures. The user sketches 2D lines of planar sections over photographs, which are automatically mapped into 3D by aligning to vanishing points or existing point geometry.

Primitive-based reconstruction. Architectural structures typically consist of an assembly of basic primitive shapes exhibiting some regularities. Schnabel et al. [SDK09] present a hole-filling algorithm that is guided by primitive detection in the point cloud. Li et al. [LWC*11] discover and optimize the global spatial relationship of geometric primitives. The interactive SmartBoxes tool proposed by [NSZ*10] utilizes the regularity of facades to quickly assemble detailed 3D primitives balancing between data fitting and structural regularity terms. Arikan et al. [ASF*13] proposed an optimization-based interactive tool that can reconstruct relatively detailed architectural models from a sparse point cloud. Lin et al. [LGZ*13] employ supervised learning to segment the urban scenes into different categories and then reconstruct 3D models using prior knowledge. The results are coarse building models that are merely approximated with a small number of textured planes. Unlike these works that target the reconstruction aspect, we focus on incorporating geometric details into the coarse models through template assembly.

Repetition detection and facade parsing. Another large body of research related to our work focuses on repetition detection and facade parsing, which aim to detect one [CZM*10, PBCL10, WFP11] or a few repetitive patterns [SHFH11, TKS*13]. The characteristics of our work that set it apart from these works include: 1) our template assembly works with a much larger number of templates and is capable of choosing and arranging a subset of the templates (quite a few templates are redundant) that best describe both the facade's content and structure (see the supplemental material). In other words, our template assembly can be described as a specialized object detection algorithm that resolves conflicts of different regular patterns caused by imperfect template matching; 2) existing facade analysis methods take a single facade image as input, while our template assembly algorithm is designed to work on multiple facades from either one building or a set of buildings.

3. Overview

Our goal is to construct detailed building models by assembling 3D templates on the faces of polygonal building mass models. Fig. 2 shows an overview of the proposed framework consisting of three main steps. The preprocessing step mainly relies on existing techniques, while the last two steps are novel algorithms introduced in this paper.

Preprocessing. The input of our framework is a set of ground level images. In our example scenes, we use between

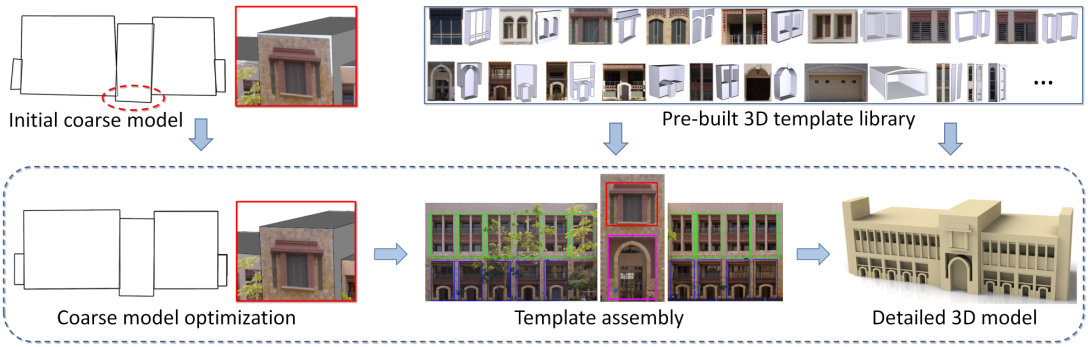


Figure 2: *An overview of our template assembly framework. Given an initial coarse model (top-left), we first enforce geometry-appearance consistency (bottom-left) through optimization. Then, the template assembly step automatically suggests appropriate 3D templates and their locations in the optimized facade texture images (bottom-middle). Finally, detailed 3D templates are transformed onto the coarse model at the suggested locations using camera parameters recovered by SfM during the preprocessing stage. This framework results in a detail rich 3D surface model (bottom-right).*

6 (Fig. 14 (a)) and approximately 200 (Fig. 1) images. We extract a 3D point cloud for these images using SfM and MVS [Wu11, WACS10]. From this point cloud, we generate a coarse approximate building mass model using a mixture of automatic and interactive tools. Alternatively, these coarse models could be generated with existing fully or semi-automated methods [ZBKB08, SSS*08, XFZ*09, VAB10, VAB12, ZN12, ZN13]. We generate 3D templates using our own interactive tool that enables a user to quickly build 3D primitives over an image. For more details on our coarse model reconstruction and the 3D template construction please refer to the supplemental materials.

Coarse model optimization. In this step, we use both geometry and appearance (texture) information to optimize the coarse model. The goal of this particular task is to improve the quality of the coarse model and its textures, by encouraging common facade structural properties (e.g., facades are piecewise planar and are either orthogonal to or parallel with each other). This can be formulated as an energy minimization problem, which inherently trades off between geometric and appearance priors, leading to more accurate and regular coarse models that have texture images consistent with their corresponding 3D geometry. Section 4 describes the details of the coarse model optimization step.

Template assembly. Since the point clouds computed using SfM and MVS lack detailed 3D information at the facade level, we rely mainly on images to assemble 3D templates on coarse models. We first detect candidate template locations in the optimized texture images of the coarse model using a template matching technique based on Histogram of Oriented Gradients features. Clearly, such matching in the image space might lead to false detections, so a further round of template candidate pruning is required for accurate and robust template assembly onto the coarse model. An appropriately sampled subset of candidate templates and their locations are chosen by solving a binary optimization problem, which trades off multiple properties of ideal template assembly including template matching score, regularity/symmetry, overall facade coverage, and sparsity. Finally, the detailed 3D templates are assembled and their corresponding geometry is added to the coarse models using transformations estimated in the coarse model construction step. Section 5 describes the details of the template assembly step.

Assumptions. In this paper, the assembly of geometric details in 3D space is reduced to choosing appropriate templates and detecting their locations in images. This restricts our framework to buildings consisting of large planar facades. We optionally make use of the Manhattan World assumption (e.g., [FCSS09, VAB10]), in the optimization to improve the coarse building mass models. Further, we assume that the set of templates available for reconstruction are taken from buildings constructed in a similar style. This assumption is reasonable for buildings in a community that are constructed by the same developer (see Fig. 3 for an example), or for buildings that stem from a city with strong coherence between facades (e.g., Haussmannian facades in Paris).

Figure 3: *Buildings within a geographical area usually have similarly detailed elements.*

4. Coarse Model Optimization

To assemble templates of facade details onto a coarse model, the facade images should first be rectified for this

purpose. The image rectification step is crucial to template assembly. Unlike conventional methods that use only image information for the rectification [LZ98], our initial coarse models are reconstructed from image sequences using SfM and MVS; thus, the camera parameters recovered from SfM can be used for the rectification. However, this rectification has some level of error due to noise in the SfM step, resulting in missing reconstruction and misalignment in the final 3D model. Our coarse model optimization is designed to enforce the geometry and appearance consistency of the coarse model, which significantly improves the final reconstruction.

We consider an initial coarse model comprised of a set of boxes $M = \{M_1, ..., M_m\}$ in 3D as seen in Fig. 4 (top). We allow this model to deform in such a way that certain geometric and appearance priors are taken into account. On the one hand, orthogonality and parallelism are quite common in architectural structures and need to be enforced in the coarse model. On the other hand, many visual cues (e.g., long straight lines corresponding to face boundaries) exist that encourage certain model deformations so as to provide a better fit to the appearance of the model's projections into each of the images it appears in.

To estimate the deformations on the initial model that incorporate the geometric and appearance priors, we formulate an energy minimization problem and allow the user to trade-off between its different terms. Unlike previous work on the topic of coarse model modification using optimization [LWC*11, ASF*13] (solely based on point clouds), we add a new regularization term that encourages consistency between the model itself and its projections into 2D images. Such a regularizer ensures a more consistent and accurate coarse model. The energy function we aim to minimize trades off between five distinct terms characterizing: geometry-appearance consistency, point cloud fidelity, face geometry, face alignment, and deviation from the initial geometry. A positive linear combination of these terms forms the energy function $E(M)$. Each of these terms is described in the following paragraphs.

The **geometry-appearance consistency term** $C(M)$ measures how well the projection of all the model's faces align with the corresponding regions in the 2D images. Note that the transformations needed for such projections have been estimated during the SfM step. We rely on geometric features of the model that manifest themselves as detectable low-level features in images. To this end, we focus on the face edges in the model, which tend to project into the images as long line segments. These line segments are detected in each image using the method of [VGJMR10]. Due to noise in the SfM process, the projections of the 3D edges might not overlay exactly on the detected line segments in the images, but instead, they tend to be quite close to each other (see the textured coarse model in Fig. 4 top-right). Therefore, we assign each face edge in the model to its closest line segments in the images. In images with camera parameters recovered by SfM, each line segment corresponds to a 3D plane that potentially generates it. Thus, we formulate this term to penalize the model by the sum of squared distances between the face edge and the 3D planes of its assigned line segments. As an effect, this term encourages the model to deform in order to satisfy the face-to-line assignment.

$$C(M) = \sum_{e \in M} dev(e), \tag{1}$$

where $dev(e)$ is the distance of a face edge e to the 3D supporting plane of the assigned line segment.

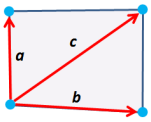
The **data fidelity term** $D(M,P)$ prevents the optimized model from deviating too much from the point cloud. It is designed to measure how well the faces of the initial coarse model M fit to their nearest 3D points $P = \{\mathbf{p}_1, ..., \mathbf{p}_n\}$.

$$D(M,P) = \sum_{(\mathbf{p},f)|dist\_point(\mathbf{p},f)<\varepsilon} dist\_point(\mathbf{p},f). \tag{2}$$

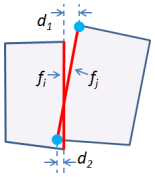
Recall that the plane segments are extracted from the point cloud using RANSAC. So here, we model $dist\_point(\mathbf{p},f)$ as the deviation of the face f from the detected plane in the point cloud. Specifically, it measures the average Euclidean distance between a vertex of the face and the detected plane. We consider only points that are ε-*close* to f (i.e., points $\mathbf{p}$ satisfying $dist\_point(\mathbf{p},f) < \varepsilon$). Throughout all our experiments, we set ε to 0.1 meters.

The **face geometry term** $F(M)$ is defined to measure how well each face in the model satisfies the co-planarity and orthogonality properties,

$$\begin{aligned} F(M) &= P(M) + O(M) \\ &= \sum_{f \in M} coplanar(f) + \sum_{f \in M} ortho(f). \end{aligned} \tag{3}$$



Here, the co-planar term $P(M)$ encourages vertices of each face to lie in the same plane. Specifically, $coplanar(f)$ is modeled as the mixed product of consecutive edge vectors and the diagonal vector of f. In our implementation, we randomly pick one of the four vertices to compute the mixed product, e.g., $\mathbf{c} \cdot (\mathbf{a} \times \mathbf{b})$ in the left example figure. The term $ortho(f)$ measures the degree of orthogonality that adjacent edges in f are, and is modeled as the sum of dot products of successive edges.



The **face alignment term** $A(M)$ discourages the existence of gaps and intersections between adjacent boxes. We only consider pairs of faces that are ε-*close* to each other and have overlap when one is projected onto the other. Let $dist\_face(f_i, f_j)$ denote the sum of distances of the vertices in f_j to the supporting plane of face f_i (e.g., $d_1 + d_2$ for the 2D example in

the left figure), and $bidist(f_i, f_j) = (dist\_face(f_i, f_j) + dist\_face(f_j, f_i))/2$, then $A(M)$ is defined as

$$A(M) = \sum_{(i,j)\in I} bidist(f_i, f_j), \tag{4}$$

where $I = \{(i, j) | \bigcap(f_i, f_j) \neq \emptyset$ and $bidist(f_i, f_j) < \varepsilon\}$, and $\bigcap(f_i, f_j) \neq \emptyset$ denotes both the projection of f_i onto f_j and the projection of f_j onto f_i are not empty.

In order to avoid drastic deviations from the user-defined initial coarse model, we add an initial geometry term $I(M)$, which regularizes the overall magnitude of the deformation. Mathematically, it measures the sum of squared distances between every vertex in M to its position in the initial coarse model.

Therefore, the underlying energy that needs to be minimized is a positive linear combination of all the aforementioned energy terms, as formulated in Equation (5).

$$\begin{aligned} E(M) =& \lambda_c C(M) + \lambda_d D(M, P) + \lambda_f F(M) + \\ & \lambda_a A(M) + \lambda_i I(M). \end{aligned} \tag{5}$$

Since it is difficult to obtain analytical forms for each of the above energy terms and their gradients, we resort to a nonlinear least-squares approach (i.e., a variant of the Levenberg-Marquardt method) to minimize the nonconvex $E(M)$ [Dev10], where the Jacobians are approximated by finite differences. In all our experiments, we use the following weights: $\lambda_c = 1.0$, $\lambda_d = 0.2$, $\lambda_f = 0.3$, $\lambda_a = 0.5$, and $\lambda_i = 0.1$. Fig. 4 shows an example optimization result. In the input coarse model (top row), the misalignment of the texture to the original model can be seen by observing the sky pixels (white and light blue) near the boundaries of the facade. Our optimization process results in a more regularized model that better aligns the facade with its texture images (bottom row).

Figure 4: *Our geometry-appearance consistent optimization resolves inconsistencies in the coarse model (e.g., intersections, gaps, and nonorthogonality) (top row), resulting in a more regularized and accurate coarse model (bottom row). Note that the geometry of the coarse model has been improved, thus a new texture image from a slightly different viewpoint is automatically chosen providing a more consistent texture for the right facade.*

5. Template Assembly

To augment the optimized coarse model with detailed geometric elements, one has to infer from the point cloud data a detailed structure of archetypal elements that tend to repeat. However, since point clouds generated by MVS tend to be sparse (compared with laser scans [NSZ*10]), noisy, and contain a large number of missing parts, performing this inference reliably and robustly is quite difficult. Therefore, we resort to a semi-automatic procedure that enables the user to explicitly define a detailed 3D template for these repeating elements from the underlying images and point cloud. To do this, we provide the user with a template construction tool, where the user can sketch template contours directly on an image. The depth of each sketched region (and equivalently its 3D position) is estimated from the point cloud. As such, the user can quickly generate a detailed 3D template (e.g., door, window, arch, etc.). Details of our interactive 3D template construction tool can be found in the supplemental material.

It is quite challenging to assemble the user-built 3D templates directly in 3D space using the information from the inherently sparse, noisy, and incomplete point clouds. Instead, in this work, we rely on images to perform this task. Specifically, by assuming facades are planar, the 3D template assembly problem is simplified to detecting the locations of the same (or very similar) objects in the rectified images. Using the camera-to-model transformations computed by SfM, we can transform these 2D detected locations back to 3D space. For this purpose, we associate each 3D template to the image region that was edited by the user to construct it (more details on the template construction are in the supplemental material). This image region is considered to be a query key image used during the 2D detection and assembly step. In what follows, we use the term *template* to refer to the 3D geometric template and its query key image interchangeably. Similarly, the term *template assembly* refers to both 3D detail enhancement of the coarse model and the assembly of 2D template query key images in the image domain.

5.1. Template descriptor and matching

To choose and locate appropriate templates that best describe each facade of the coarse model, we require template matching techniques that are capable of capturing the underlying geometric properties of the projected 3D elements (i.e., the contours). Based on the state-of-the-art object detection literature, the Histogram of Oriented Gradients (HOG) feature has been proven to be efficient, effective, and robust for this task. This is attributed to the fact that local object appearance and shape can be characterized well by the distribution of local intensity gradients or edge directions [DT05]. In this paper, we use the HOG feature to describe our templates and potential detection locations in the image domain.

Recall that the coarse models have been optimized in Section 4 alleviating the effect of perspective distortion in the texture images. We define the score $S(\mathbf{T},\mathbf{F})$ for matching a template query key image $\mathbf{T}$ with a facade texture image region $\mathbf{F}$ as the Normalized Cross Correlation (NCC) of their HOG descriptors. Let $\mathbf{I}$ denote the texture image of the facade obtained from the geometry-appearance consistent optimization step in Section 4, and $\mathbf{T} = \{\mathbf{t_1}, ...\mathbf{t_k}\}$ denote the k templates' query key images. We initially perform HOG-based template matching for each query key image in $\mathbf{T}$. The results are a set of candidate locations for each individual template (Fig. 5). Note that the initial locations of different templates may overlap due to the absence of high-level constraints (e.g., regularities and alignments).

Figure 5: *Candidate template locations detected by the HOG-based template matching algorithm. Different colors indicate different templates. The template images are shown on the right. Note, these templates may come from the facades of other buildings.*

One possible way to resolve this overlap problem is by performing the detection in a greedy manner, e.g., only the template that has the highest $S(\mathbf{T},\mathbf{F})$ score is maintained at a particular location. However, due to illumination variations, occlusions, and the complexity of facade structures, the highest score does not always suggest the best template. This is typically true when the number of templates is large or when occlusions and varying illumination conditions persist.

In our work, we exploit high-level structural properties (i.e. repetitions and alignments) of facades to alleviate these overlap ambiguities. Compared to [MZWG07] that makes stronger assumptions on facade regularity (a single grid of elements), we are also interested in handling interleaved grids (e.g., Figures 4 and 6 in Supplemental Material 2 are two such examples). First they compute the location of individual template elements using translational symmetries and edge detection. Next, they find suitable templates for given fixed rectangles in an image. By contrast, we jointly optimize for template positions and template types by formulating a discrete (binary) optimization problem that incorporates three interacting terms: 1) a template matching cost, 2) a facade regularity cost, and 3) a spatial coverage cost. The optimal assembly that describes the facade is one of the many potential groupings of individual template detections. In the following, we first describe the generation of these candidate groups, followed by our formulation of the discrete optimization.

Figure 6: *Detected template locations for a given template.*

5.2. Candidate group generation

We first generate a restricted power set of potential groupings of the initial template matching detections, denoted by $G = \{g_1, ...g_n\}$. Thus, the optimal representation of the facade is a subset of the potential groups that satisfy some requirements formulated as the aforementioned three energy terms. Here, we only consider potential groups whose elements are spatially adjacent. For example in Fig.6, the candidate groups are $\{A\}$,...,$\{H\}$, $\{A,B\}$,...,$\{F,G\}$,$\{C,H\}$,..., and $\{A,B,C,D,E,F,G,H\}$.

Let $\mathbf{X}$ denote the binary labels for all the candidate groups. Consequently, the solution to the template assembly problem is a subset of non-overlapping candidate groups that minimizes the following three costs:

The **template matching cost term** $E_T(\mathbf{X})$ measures the overall matching confidence for the entire facade. It is defined as how far it is from a perfect match, where the score of a perfect match using NCC of the HOG descriptor is 1.

$$E_T(\mathbf{X}) = \sum_{i=1}^{n} [(1 - s_i) \cdot x_i]. \tag{6}$$

Here, s_i is the average matching score of all elements in group g_i, and x_i denotes the binary labels of g_i, where 1 indicates *chosen* and 0 indicates *not chosen*.

The **regularity term** $E_R(\mathbf{X})$ measures the overall regularity of the assembly,

$$E_R(\mathbf{X}) = \sum_{i=1}^{n} (r_i \cdot x_i). \tag{7}$$

Here, the regularity r_i of a group g_i is measured as how far g_i is from a perfectly regular pattern. It is defined as the maximum variance of interval and alignment of g_i.

$$r_i = \begin{cases} \max(\frac{var_I(g_i)}{w_i}, \frac{var_A(g_i)}{h_i}), & \text{for a horizontal group} \\ \max(\frac{var_I(g_i)}{h_i}, \frac{var_A(g_i)}{w_i}), & \text{for a vertical group,} \end{cases} \tag{8}$$

where w_i and h_i are the average width and height of the elements in g_i. $var_I(g_i)$ and $var_A(g_i)$ are the interval variance and alignment variance of g_i.

The **coverage term** $E_C(\mathbf{X})$ encourages that more of the facade image be covered by candidate groups.

$$E_C(\mathbf{X}) = 1 - \frac{\sum_{i=1}^{n} [area(g_i) \cdot x_i]}{area(\mathbf{I})}. \tag{9}$$

To prevent multiple overlapping groups from contributing to this cost, later we will add an exclusion constraint that allows one group maximum to be chosen at any given location in the image.

5.3. Assembly optimization

The template assembly optimization problem can be formulated as finding an optimal labeling for each candidate group so as to minimize a weighted sum of the above three terms. Before we describe how template assembly is optimized, we make the following observation: since generally a facade can be represented by a number of subsets of individual template groups, we seek to find the most compact among these representations (i.e., the smallest number of template groups). Inspired by recent work in sparse representation [CT06, WYG*09], we encourage the structure of the facade to be represented by a minimum number of candidate groups. Therefore, we add a sparsity-inducing regularization term $E_S(\mathbf{X})$ to the objective function.

$$E_S(\mathbf{X}) = \sum_{i=1}^{n} x_i. \tag{10}$$

The overall objective function to be minimized is

$$E(\mathbf{X}) = \lambda_T E_T(\mathbf{X}) + \lambda_R E_R(\mathbf{X}) + \lambda_C E_C(\mathbf{X}) + E_S(\mathbf{X}). \tag{11}$$

Thus, our optimization problem can be formulated as

$$\begin{aligned} \min_{\mathbf{X}} \quad & E(\mathbf{X}) \\ \text{s.t.} \quad & x_i + x_j \leq 1 \quad \forall g_i \bigcap g_j \neq \emptyset, \qquad 1 \leq i, j \leq n \\ & x_i \in \{0,1\}, \qquad\qquad\qquad\quad 1 \leq i \leq n, \end{aligned} \tag{12}$$

where $x_i + x_j \leq 1$ is an exclusion constraint, which ensures that one group maximum among a set of overlapping groups is chosen at a given location in the image. λ_T, λ_R, and λ_C are weights that balance between the template matching, regularity, and spatial coverage costs. In all our experiments, we use the following weights: λ_T = 60, λ_R = 200, and λ_C = 970. These weights are learned from a set of ground truth assemblies using linear regression [HTF*09].

Our formulation of the template assembly results in a 0-1 (binary) linear programming problem. We solve it using the conventional Gurobi solver [Gur], which is quite efficient with a runtime of 3.01 seconds for an assembly problem of 961 candidate groups and 93,944 non-overlapping constraints. Fig. 7 (top) shows the assembly result of the facade in Fig. 5. As a comparison, we also show the assembly of the same templates for this facade using the greedy approach discussed earlier in this section (bottom).

5.4. Detail enhancement

The previous template assembly optimization step results in a compact representation of the facade texture images.

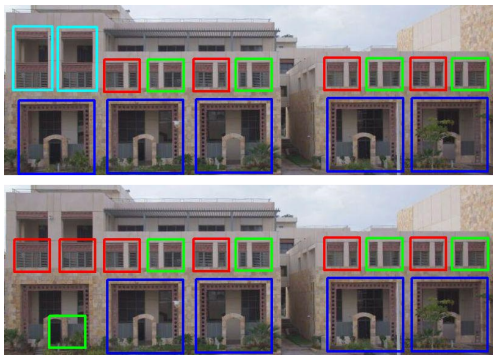


Figure 7: *Template assembly results using our optimization-based approach (top) as compared to results from the greedy approach (bottom).*

To enrich the detail of the 3D coarse model, we use the optimal location of each detected template to backproject its 3D template onto the 3D face in the coarse model. This backprojection process makes use of the camera-to-model transformations estimated by SfM in the aforementioned preprocessing stage. The geometry of the 3D templates is appropriately transformed onto the optimized 3D locations in the corresponding facade face. The result is a 3D mesh model of the building enriched with fine-grained geometric details.

6. Results and Discussion

We have applied our template assembly algorithm on a variety of architectural scenes.

Coarse model optimization results. First, we would like to present and discuss the necessity of the coarse model optimization step. Fig. 4 shows how our geometry-appearance consistent optimization resolves inconsistencies (i.e., intersections, gaps, and nonorthogonality). As a result, the texture quality has been significantly improved. To quantitatively evaluate the texture improvement, we use the area of texture overlap with ground truth divided by their union as a measure. Texture quality of the three facades in the model (before/after optimization) are 94.2%/98.3%, 93.5%/99.1%, and 91.4%/97.7% respectively. This process is crucial for the latter template assembly step. In Fig. 8, we show the effect of the coarse model optimization step on the 2D template assembly and the final 3D reconstruction.

Template assembly results. After consistency between geometry and appearance is enforced, we then perform template assembly in the optimized texture images, followed by detail enhancement for the coarse models. As can be seen from Figures 1, 11, 12, 13, and 14, our optimization-based template assembly algorithm is quite robust to varying illumination and occlusions, resulting in promising detailed 3D reconstructions.

Fig. 1 shows two buildings reconstructed with details using our template assembly approach. We also show the textured version of the reconstructed model in Fig. 10. The realism has been significantly improved by adding more details into the coarse model. In Fig. 11, the building is from the same region as the previous example shown in Fig. 1. Although these two buildings have different mass models, they share the same detailed elements. Therefore, the details for this building are correctly recovered even though the templates are extracted from the one shown in Fig. 1.

Fig. 12 shows another set of buildings reconstructed using our template assembly method. Although the facades in Fig. 12 (a) are occluded by trees, our algorithm successfully localizes most of the template instances. This is possible because our assembly optimization computes a compromise between matching scores of individual templates and pattern regularity. A door in the middle of the first floor is missed due to severe occlusions by trees. In Fig. 12(b), the facades exhibit some irregularities (different adjacent windows), nevertheless, our method still produces good assembly results.

In Fig. 13 four buildings from another area feature completely different structures and appearances. Thus with our interactive template construction tool, we build another set of templates to perform our automatic template assembly algorithm.

Fig. 14 shows several street-side facades reconstructed with details. Although the windows in the middle facade are quite similar in style and appearance, our template assembly algorithm succeeds in choosing appropriate templates and detecting the locations for most of them.

Comparison. In addition to the 3D examples discussed above, we also tested our 2D template assembly algorithm on a variety of facade images, and conducted a comprehensive comparison of our optimization-based assembly algorithm with two other approaches: 1) greedy - a greedy method where we iteratively add the templates with the highest matching score. This method is identical to our framework, except that we replace the template assembly step with a greedy search. 2) mutual information - template matching using *Mutual Information* (as suggested by [MZWG07]) followed by our assembly optimization step. For the visual comparison please refer to the supplementary material. In Fig. 9 we show a quantitative comparison between the three methods. We manually marked 24 facades with ground truth labels and computed an assembly score as the area of overlap with ground truth divided by their union. On average, our method achieves 93% overlap compared with 85% for greedy and 51% for mutual information. The comparison shows that the greedy method is prone to produce false template instances and that Mutual Information is not competitive in detecting the correct template locations, while our optimization-based method performs the best among the three.

Timings. Total reconstruction time for each of our scenes is less than 5 minutes. As can be seen from Tab. 1, adding details to the coarse models is much faster than the coarse model construction step. In Fig. 11, since all templates were built for the scene shown in Fig. 1, it takes less than 4 minutes (including the coarse model construction, coarse model optimization, and template assembly) to obtain this detailed 3D model. In Fig. 13 the reconstruction time for all four detailed buildings is less than 5 minutes.

Table 1: *Detailed summary of coarse model sizes (quantified by the number of boxes), detailed model sizes (number of faces), coarse model reconstruction runtime, and template assembly runtime for the 3D examples presented in this paper.*

Figure	Coarse (# box)	Detail (# face)	Coarse model	Template assembly
1	18	4,897	4 min	34 sec
11	18	4,479	3 min	28 sec
12 (a)	5	2,754	1 min	39 sec
12 (b)	7	2,661	2 min	32 sec
12 (c)	7	2,856	1 min	26 sec
12 (d)	11	5,330	2 min	47 sec
13 (a)	4	938	1 min	11 sec
13 (b)	4	942	1 min	14 sec
13 (c)	5	1,046	1 min	16 sec
13 (d)	5	1,046	1 min	15 sec
14	7	3,225	1 min	67 sec

Limitations. Our template assembly-based reconstruction strategy is suitable for reconstructing a set of buildings that consists of the same or similar detailed structures (i.e., elements repeating among the target buildings or facades). In some instances, this is an advantage of our template assembly framework; however, when repetitions do not exist, the templates must first be identified using manual annotation.

With the assumption that facades are planar, the assembly of detailed templates in 3D space is simplified to choosing appropriate templates and detecting their locations in the image domain. This restricts our framework's applicability to buildings consisting of planar facades.

Our template assembly is designed on top of the robust HOG-based template matching algorithm. Although the HOG feature is not invariant to projective distortions and occlusions caused by protruding components, we observe that it performs well in all our experiments. This is because the template and the facade images have similar projective distortions and occlusions (taken from similar view points); this is typically true for low-rise buildings. It may fail in extreme situations, such as a template of a balcony from the first floor getting matched to balconies on higher floors.

We currently do not have a regularization term to

complete missing parts of the facade. For example, the door in Fig. 12 (a) (left facade) is almost completely occluded by trees. In future work, we would like to extend our template assembly framework so that plausible templates are suggested for occluded regions. One useful ingredient might be the explicit detection of vegetation in the images.

7. Conclusions and Future Work

We present an efficient urban reconstruction framework based on an effective template assembly algorithm. Our algorithm relies on image information and reusable detailed 3D templates to enrich the 3D models. We use image information in two key steps of the reconstruction pipeline. First, we use it to improve the geometry-appearance consistency of the coarse mass models. Second, we use it to detect templates on the facades of the coarse mass models. We incorporate prior knowledge about facades into our optimization process for both steps, resulting in an effective urban reconstruction framework.

In future work we plan to extract templates from laser scans and match them to images. We would also like to investigate the detection of occlusions in facade images.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions. Special thanks goes to Dr. Neil Smith for providing us the easy-to-use data acquisition system and the data for initial experiments. We also thank Guangfan Pan for texturing the model shown in Fig. 10. This work was supported by the KAUST Visual Computing Center.

References

[ASF*13] ARIKAN M., SCHWÄRZLER M., FLÖRY S., WIMMER M., MAIERHOFER S.: O-snap: Optimization-based snapping for modeling architecture. *ACM Transactions on Graphics 32*, 1 (2013), 6. 2, 4

[CMZP14] CEYLAN D., MITRA N. J., ZHENG Y., PAULY M.: Coupled structure-from-motion and 3d symmetry detection for urban facades. *ACM Transactions on Graphics 33*, 1 (2014), 2. 2

[CT06] CANDES E. J., TAO T.: Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on 52*, 12 (2006), 5406–5425. 7

[CZM*10] CHENG M.-M., ZHANG F.-L., MITRA N. J., HUANG X., HU S.-M.: Repfinder: Finding approximately repeated scene elements for image editing. *SIGGRAPH 29*, 4 (2010), 83:1–8. 2

[Dev10] DEVERNAY F.: C++ minpack. http://devernay.free.fr/hacks/cminpack/, 2010. 5

[DT05] DALAL N., TRIGGS B.: Histograms of oriented gradients for human detection. In *CVPR* (2005), vol. 1, pp. 886–893. 5

[DTC04] DICK A. R., TORR P. H. S., CIPOLLA R.: Modelling and interpretation of architecture from several images. *Int. J. Comput. Vision 60*, 2 (2004), 111–134. 2

[FCSS09] FURUKAWA Y., CURLESS B., SEITZ S. M., SZELISKI R.: Manhattan-world stereo. In *CVPR* (2009), pp. 1422–1429. 3

[FP10] FURUKAWA Y., PONCE J.: Accurate, dense, and robust multi-view stereopsis. *PAMI 32*, 8 (2010), 1362–1376. 1, 2

[GSC*07] GOESELE M., SNAVELY N., CURLESS B., HOPPE H., SEITZ S.: Multi-view stereo for community photo collections. In *ICCV* (2007), pp. 1–8. 2

[Gur] GUROBI: Gurobi optimization. http://www.gurobi.com/. 7

[HTF*09] HASTIE T., TIBSHIRANI R., FRIEDMAN J., HASTIE T., FRIEDMAN J., TIBSHIRANI R.: *The elements of statistical learning*, vol. 2. Springer, 2009. 7

[HZ00] HARTLEY R., ZISSERMAN A.: *Multiple view geometry in computer vision*, vol. 2. Cambridge Univ Press, 2000. 1

[LGZ*13] LIN H., GAO J., ZHOU Y., LU G., YE M., ZHANG C., LIU L., YANG R.: Semantic decomposition and reconstruction of residential scenes from lidar data. *SIGGRAPH 32*, 4 (2013). 1, 2

[LWC*11] LI Y., WU X., CHRYSATHOU Y., SHARF A., COHEN-OR D., MITRA N. J.: Globfit: consistently fitting primitives by discovering global relations. In *ACM Transactions on Graphics* (2011), vol. 30, ACM, p. 52. 2, 4

[LZ98] LIEBOWITZ D., ZISSERMAN A.: Metric rectification for perspective images of planes. In *CVPR* (1998), pp. 482–488. 4

[MWA*13] MUSIALSKI P., WONKA P., ALIAGA D. G., WIMMER M., GOOL L., PURGATHOFER W.: A survey of urban reconstruction. In *Computer Graphics Forum* (2013). 2

[MZWG07] MÜLLER P., ZENG G., WONKA P., GOOL L. J. V.: Image-based procedural modeling of facades. *ACM Transactions on Graphics 26*, 3 (2007), 85. 6, 8, 10

[NSZ*10] NAN L., SHARF A., ZHANG H., COHEN-OR D., CHEN B.: Smartboxes for interactive urban reconstruction. *SIGGRAPH* (2010). 2, 5

[PBCL10] PARK M., BROCKLEHURST K., COLLINS R. T., LIU Y.: Translation-symmetry-based perceptual grouping with applications to urban scenes. In *ACCV*. 2010. 2

[PF08] POLLEFEYS M. NISTER D., FRAHM E.: Detailed real-time urban 3D reconstruction from video. *Int. J. Comput. Vision 78*, 2-3 (2008), 143–167. 2

[SDK09] SCHNABEL R., DEGENER P., KLEIN R.: Completion and reconstruction with primitive shapes. *EUROGRAPHICS 28*, 2 (2009), 503–512. 2

[SHFH11] SHEN C.-H., HUANG S.-S., FU H., HU S.-M.: Adaptive partitioning of urban facades. In *ACM Transactions on Graphics* (2011), vol. 30, p. 184. 2

[SSS*08] SINHA S. N., STEEDLY D., SZELISKI R., AGRAWALA M., POLLEFEYS M.: Interactive 3D architectural modeling from unordered photo collections. *ACM Transactions on Graphics 27*, 5 (2008), 1–10. 2, 3

[TKS*13] TEBOUL O., KOKKINOS I., SIMON L., KOUTSOURAKIS P., PARAGIOS N.: Parsing facades with shape grammars and reinforcement learning. *PAMI 35*, 7 (2013), 1744–1756. 2

[VAB10] VANEGAS C. A., ALIAGA D. G., BENES B.: Building reconstruction using manhattan-world grammars. In *CVPR* (2010), pp. 358–365. 1, 3

[VAB12] VANEGAS C. A., ALIAGA D. G., BENES B.: Automatic extraction of manhattan-world building masses from 3d laser range scans. *Visualization and Computer Graphics, IEEE Transactions on 18*, 10 (2012), 1627–1637. 1, 3

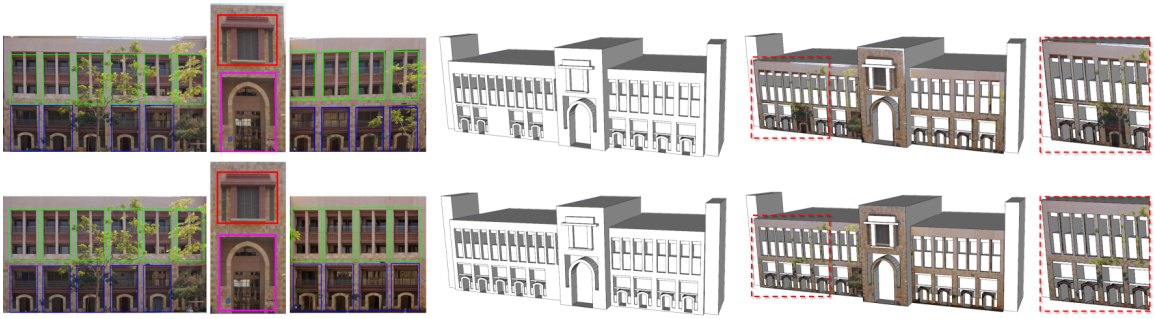


Figure 8: *A comparison of template assembly results without (top) and with (bottom) the geometry-appearance optimization of the coarse model. From left to right: the template assembly result, the final 3D reconstruction, the transformed 3D templates overlayed on the textured coarse model, and a close-up view. Clearly, the geometry-appearance optimization step leads to more consistent geometry, better localized template detections, and a more accurate detailed model.*

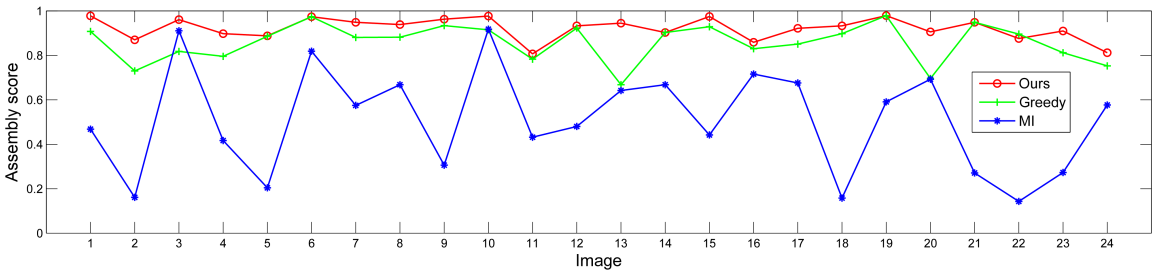


Figure 9: *A quantitative comparison of our optimization-based template assembly with two competing methods: greedy and mutual information [MZWG07]. We applied these methods to all 24 facades in our data set. The assembly score is measured as the ratio of the overlap area between the automatically detected templates and the manually marked ground truth to their union. A ratio closest to 1 indicates the best results.*

Figure 10: *Textured detailed models of the two buildings in Fig. 1. The inset shows the real photograph for a visual comparison.*

[VAW*09] VANEGAS C. A., ALIAGA D. G., WONKA P., MUELLER P., WADDELL P., WATSON B.: Modeling the appearance and behavior of urban spaces. In *Proc. of Eurographics State-of-the-Art Report* (2009). 2

[VGJMR10] VON GIOI R. G., JAKUBOWICZ J., MOREL J.-M., RANDALL G.: Lsd: A fast line segment detector with a false detection control. *PAMI 32*, 4 (2010), 722–732. 4

[WACS10] WU C., AGARWAL S., CURLESS B., SEITZ S. M.: Multicore bundle adjustment. In *CVPR* (2010), pp. 3057–3064. 2, 3

[WACS12] WU C., AGARWAL S., CURLESS B., SEITZ S. M.: Schematic surface reconstruction. In *CVPR* (2012), pp. 1498–1505. 2

[WFP11] WU C., FRAHM J.-M., POLLEFEYS M.: Repetition-based dense single-view reconstruction. In *CVPR* (2011), pp. 3113–3120. 2

[Wu11] WU C.: Visualsfm: A visual structure from motion system. http://ccwu.me/vsfm/. 2, 3

[WYG*09] WRIGHT J., YANG A. Y., GANESH A., SASTRY

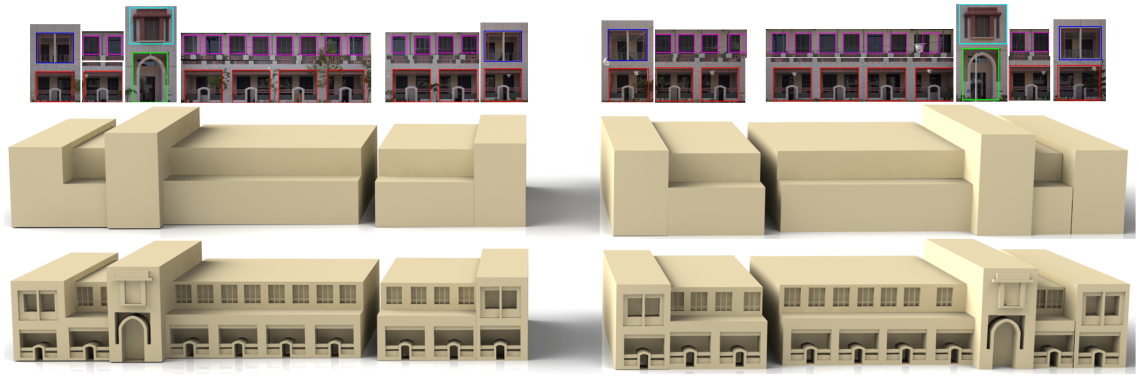


Figure 11: *Two different buildings with the same style as those in Fig. 1 are reconstructed with details using the same set of 3D templates.*

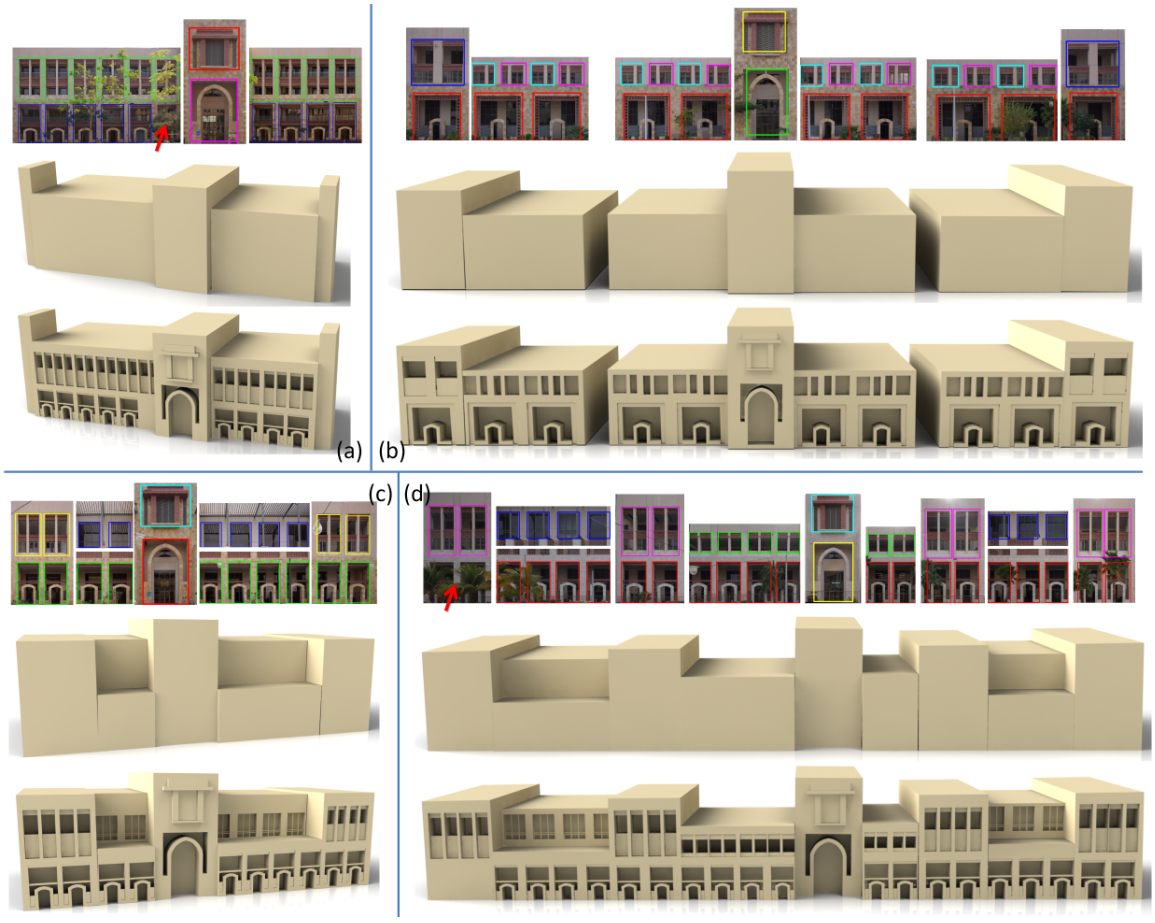


Figure 12: *A set of buildings reconstructed with details by our template assembly method using the same set of 3D templates. Due to severe occlusions, the doors marked by red arrows are not detected in the template matching step, thus, no 3D reconstruction exists for them.*

S. S., MA Y.: Robust face recognition via sparse representation. *PAMI 31*, 2 (2009), 210–227. 7

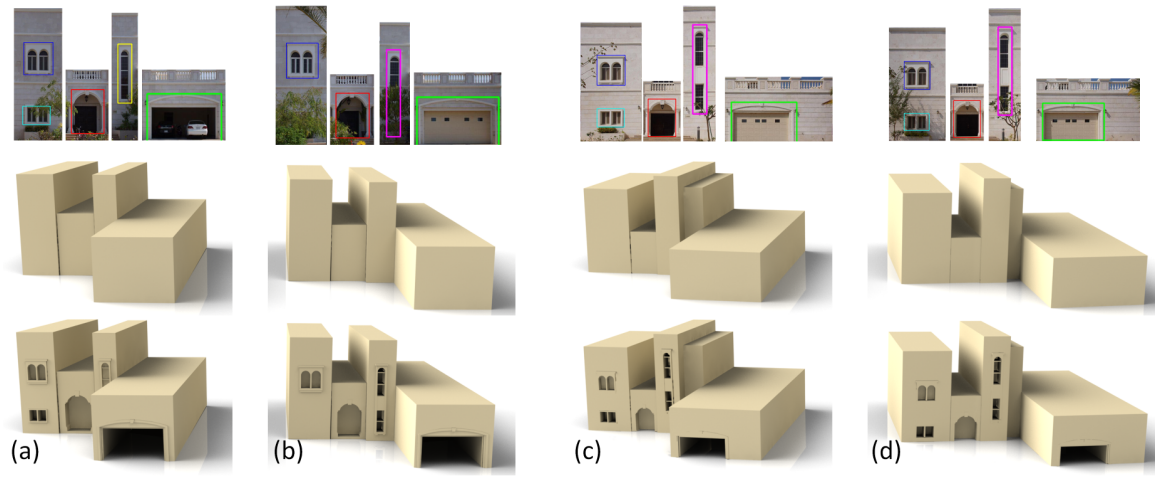


Figure 13: *Detailed reconstruction of another set of buildings with a different architectural style.*

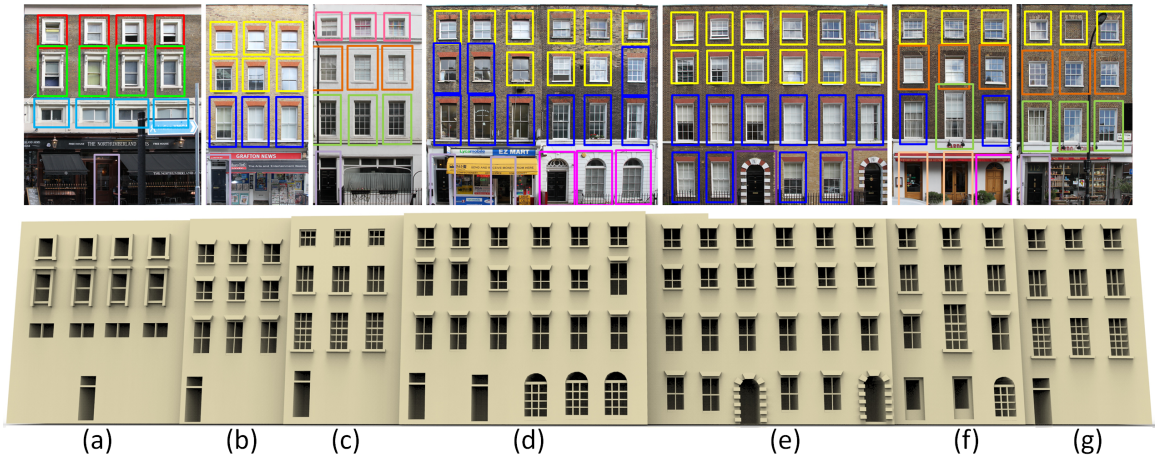


Figure 14: *Detailed reconstruction of seven street-side facades.*

[WYJT10] WU T.-P., YEUNG S.-K., JIA J., TANG C.-K.: Quasi-dense 3d reconstruction using tensor-based multiview stereo. In *CVPR* (2010), IEEE, pp. 1482–1489. 2

[WZ02] WERNER T., ZISSERMAN A.: New techniques for automated architecture reconstruction from photographs. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark* (2002), vol. 2, pp. 541–555. 2

[XFT*08] XIAO J., FANG T., TAN P., ZHAO P., OFEK E., QUAN L.: Image-based facade modeling. In *ACM Transactions on Graphics* (2008), vol. 27, p. 161. 2

[XFZ*09] XIAO J., FANG T., ZHAO P., LHUILLIER M., QUAN L.: Image-based street-side city modeling. *ACM Transactions on Graphics 28*, 5 (2009), 114:1–114:12. 1, 2, 3

[ZBKB08] ZEBEDIN L., BAUER J., KARNER K., BISCHOF H.: Fusion of feature-and area-based information for urban buildings modeling from aerial imagery. In *ECCV*. 2008, pp. 873–886. 3

[ZN12] ZHOU Q.-Y., NEUMANN U.: 2.5 d building modeling by discovering global regularities. In *CVPR* (2012), pp. 326–333. 1, 3

[ZN13] ZHOU Q.-Y., NEUMANN U.: Complete residential urban area reconstruction from dense aerial lidar point clouds. *Graphical Models 75*, 3 (2013), 118–125. 1, 3