

Image-based Building Regularization Using Structural Linear Features

Jinglu Wang, Tian Fang, Qingkun Su, Siyu Zhu, Jingbo Liu, Shengnan Cai, Chiew-Lan Tai, and Long Quan

Abstract—Reconstructed building models using stereo-based methods inevitably suffer from noise, leading to the lack of regularity which is characterized by straightness of structural linear features and smoothness of homogeneous regions. We leverage the structural linear features embedded in the mesh to construct a novel **surface scaffold structure** for model regularization. The regularization comprises two iterative stages: (1) the linear features are semi-automatically proposed from images by exploiting photometric and geometric clues jointly; (2) the scaffold topology represented by spatial relations among the linear features is optimized according to data fidelity and topological rules, then the mesh is refined by adjusting itself to the consolidated scaffold. Our method has two advantages. First, the proposed scaffold representation is able to concisely describe semantic building structures. Second, the scaffold structure is embedded in the mesh, which can preserve the mesh connectivity and avoid stitching or intersecting surfaces in challenging cases. We demonstrate that our method can enhance structural characteristics and suppress irregularities in the building models robustly in some challenging datasets. Moreover, the regularization can significantly improve the results of general applications such as simplification and non-photorealistic rendering.

Index Terms—I.3.5.f Modeling packages, I.4.5. Reconstruction.

1 INTRODUCTION

MULTI-VIEW 3D reconstruction of large-scale urban scenes finds its practical applications in virtual reality, urban planning and 3D mapping, and has also been promoted by various approaches. Traditional stereo-based methods [1], [4], [5], [6] can generate high-resolution mesh models, but they are inevitably laden with a lot of noise, distorted structures and incomplete geometry due to imperfect camera geometry and object occlusion. Therefore, regularizing reconstructed models to be clean and high-quality ones is strongly desired, which can significantly improve the subsequent texture mapping, rendering and data compression.

Automatic regularization methods either treat meshes as general signals to apply local mesh processing [7], [8], [9], or fit simplistic 3D primitives to 3D point clouds [3], [10], [11]. However, the features of high-frequency, namely the mesh details, are difficult to preserve in the presence of noise, as seen in Fig. 1g.

Others refer to semi-automatic methods [12], [13], [14], [15] which bring users' intelligence into the modeling loop. The users identify primitive features and structures in images to resemble the discontinuities on 3D surface. However, the main inconvenience of these interactive methods is that the users are usually required to specify complex topological clues via interventions, such as linking separate edges or drawing closed polygons. Thus it is difficult to distribute large-scale tasks to untrained users.

Inspired by the fact that straightness of 3D linear features

and smoothness of bounded regions are highly representative of the building regularity, we propose a novel *surface scaffold structure*, which consists of a set of structural 3D linear features (Fig. 1d) to achieve the building regularization. First, we back-project 2D image lines onto the input mesh to obtain the initial linear features which comprise sequences of mesh edges. Such linear features may be in inaccurate poses (positions and orientations) due to mesh noise and registration errors. The consolidation process can be employed to refine the initial scaffold structure. There are two steps in the process, namely, the scaffold topology optimization and the global shape refinement. The former step involves deriving correct incidence and parallelism relations among nearby scaffold lines rather than only inferring incidence relations of lines using the hard constraint on proximity of endpoints [16]. The latter step involves refining the poses of the linear features to satisfy the derived topology in a least-square manner and smoothing the other mesh parts. Only the linear features are forced to be straight since they correspond to sharp building edges intuitively, while the bounded mesh regions are not restricted to be planar but smooth. Thus, severe distortion will not be produced and the original mesh topology is retained.

Proposing the linear features from images is challenging. Traditional 2D line detection in real images tends to extract fragmented and spurious lines [16], [17], [18], resulting in uncertainty of the positions of the endpoints. We apply an efficient approach to extract accurate and complete 2D lines by selecting an appropriate image scale, and then construct a binary classifier to filter out the lines that are not structural or of erroneous 2D-to-3D correspondences by utilizing 2D and 3D information jointly. Also, human perception is used to facilitate the line proposal and furthermore, the strong urban prior, symmetry, is exploited to reduce the user intervention. Interactively, the user can draw a line

• All authors are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.
E-mail: {jwangae, tianft, qsuaa, szhu, jingbo, scaiad, taicl, quan}@cse.ust.hk

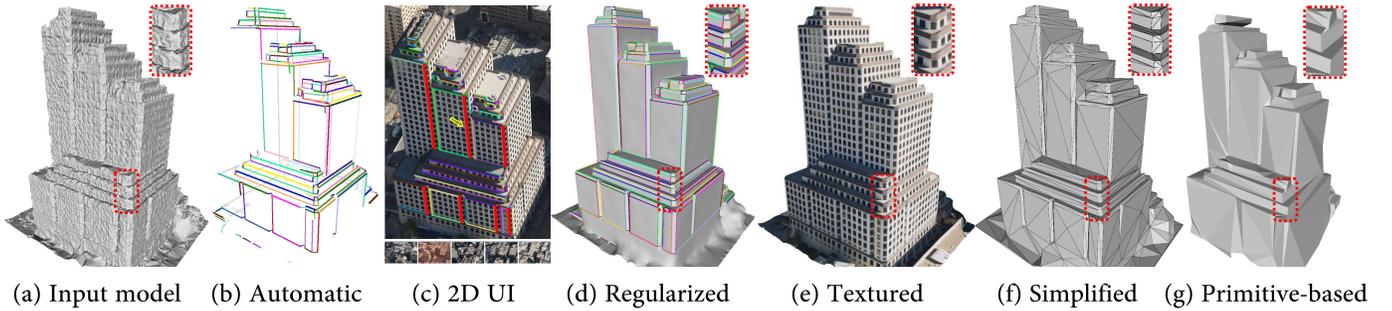


Fig. 1. Image-based Building Scaffolding. (a) Input mesh from multi-view reconstruction [1]. (b) Consolidated scaffold with the automatically proposed scaffold lines. (c) 2D user interface. The automatically selected views for proposing scaffold lines are shown at the bottom. In an interactive iteration, the user draws a line (the red line indicated by the yellow arrow), and then the system proposes symmetric lines (the other red lines) based on the existing scaffold. (d) Regularized mesh overlaid with the finally consolidated scaffold. (e) Textured regularized model showing the pleasing texture mapping thanks to its accurate geometry. (f) Simplification [2] result of our regularized model. Fine details are well preserved as shown in the close-up. (g) Results of the primitive-based method (Chauve et al. [3]) which fail to capture the fine details and the intersection boundaries of planes precisely. The facet number is the same as the model in (f).

in an image, then more lines symmetric to the user input can be automatically proposed (Fig. 1c). Besides, the images for proposing linear features are automatically selected to improve the efficiency by simplifying a view redundancy graph.

In summary, the building regularization is achieved based on the surface scaffold structure by iteratively performing 1) automatic/interactive proposal of the anticipated linear features from the selected images, and 2) optimization of the scaffold topology and refinement of the global mesh. Further applications, such as mesh simplification and non-photorealistic rendering, can demonstrate the superiority of our regularization method over other related methods. The contributions of this paper are

- A novel scaffold structure that can concisely describe the building structure and maintain the model regularization.
- An noise-insensitive scaffold structure consolidation method of deriving the correct spatial relations among 3D linear features and refining the surface shape according to the derived topology.
- The automatic/interactive line proposal by utilizing photometric, geometric information, and urban priors jointly.

1.1 Related Work

Automatic methods. Reconstructed elements from multi-view images of most methods [1], [19], [20] are depth maps, voxels, or 3D points, while Vu et al. [6] start directly from a rough mesh and further refine it according to a photo consistency energy, benefiting from the connectivity and visibility information of the reconstruction. To enhance the regularity of the reconstruction, a few recent works address some primitive-based methods [21], [22], [23], [24], which fit a set of geometric primitives to input data, and then adequately assemble them into a surface. However, these methods only handle data that contains a small proportion of noise, e.g., laser scans, synthetic point clouds, or reconstructions of simple scenes. Though Chauve et al. [3] can reconstruct piecewise planar meshes from noisier data by adaptively decomposing the 3D space induced by the

fitted planes, like most primitive-based methods that strictly rely on the perfection of the primitives detection, distorted structures may be produced when stitching or intersecting the primitives by any under-fitting or over-fitting. Moreover, since the primitive representation is highly parameter-dependent, it is difficult to adapt to large-scale scenes.

Interactive methods. Early work [14] models architectures with intensive user assistance. The user first needs to manually select a small number of images, mark edges in images and correspond the image edges to the model edges. Recently, Sinha et al. [12] use structure from motion results and image edges to interactively create piecewise planar models of architectures. The user is asked to sketch 2D outlines of planar sections overlaid on images. Their system still requires users to specify complex topology of the lines and planes. Arikan et al. [25] enable users to refine the fitted planes by snapping the polygons together. In our interface, the user only needs to draw unorganized lines and our system is able to automatically optimize the topology. In addition, symmetry is interleaved to alleviate the user intervention.

Topological constraint based methods. Some works improve the reconstruction by incorporating topological constraints of linear features. Recent works [26], [27] extract structures of objects from line drawings with derived topology for surface reconstruction, but they only employ geometric cues without involving photometric information, which depend strictly on the precision of the input geometry. The image-based method [16] reconstructs 3D lines from multi-view images by imposing incidence constraints on the reconstruction. Some other methods also infer the parallelism of the 3D lines by exploiting the cues of vanishing point in images. The authors of [18] reconstruct 3D lines in dominant planes on architectural scenes to recover piecewise planar depth maps. Ramalingam et al. [17] utilize the Manhattan assumption to extract 3D lines in three orthogonal directions. These methods either considering the incidence relation only according to position proximity or restricting parallel 3D lines to be in dominant directions often discard the evidence provided by short lines or lines not in dominant directions. Consequently, some distinct lines in close proximity are often merged, and parallelism

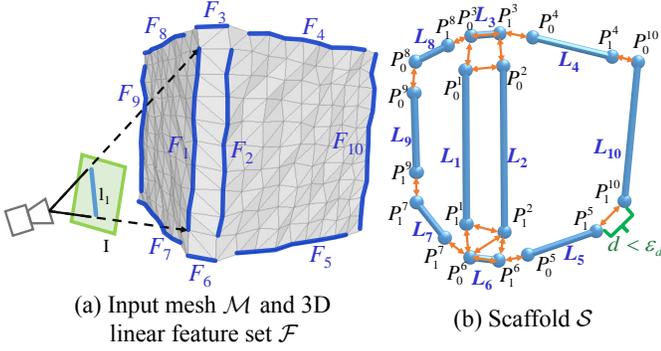


Fig. 2. The surface scaffold structure, which consists of a set of 3D linear features \mathcal{F} (shown in blue) on the mesh \mathcal{M} (a) and their corresponding scaffold lines L_i in the scaffold \mathcal{S} (b). The linear feature F_1 comprising a sequence of mesh edges is obtained by back-projecting an image line l_1 in view I onto \mathcal{M} . Two endpoints of the scaffold lines are connected by an orange arrow if their distance d is smaller than ϵ_d . (Only visible linear features are shown here.)

not related to dominant directions is neglected.

The rest of the paper is organized as follows. Section 2 presents the representation for the surface scaffold structure and its topology. Section 3 introduces the automatic method for proposing lines. The scaffold structure consolidation is detailed in Section 4, including the scaffold topology optimization and global surface refinement. Section 5 describes the interactive line proposal and the 2D-3D user interface. Experimental results are presented and evaluated in Section 6.

2 SCAFFOLDING

2.1 Surface Scaffold Structure

The key idea of our method is to use a set of structural linear features on the surface to capture and enhance the building regularity. Given an input mesh generated from any multi-view 3D reconstruction method, $\mathcal{M} = (\mathcal{V}_m, \mathcal{E}_m)$, with vertex set \mathcal{V}_m and edge set \mathcal{E}_m , we introduce the *surface scaffold structure* to refine the surface shape and thereby achieve model regularization. As illustrated in Fig. 2, the surface scaffold structure contains

- 3D linear feature set $\mathcal{F} = \{F_i\}, i \in [1, N]$, where F_i is a mesh line comprising a sequence of connected edges on \mathcal{M} . Each mesh line F_i is obtained by back-projecting an image line l_i onto \mathcal{M} (Fig. 2a).
- Scaffold \mathcal{S} organizes a set of scaffold lines $\mathcal{L}_s = \{L_i\}, i \in [1, N]$, and defines a scaffold graph which represents the topology of the endpoints of the scaffold lines. A scaffold line L_i is a 3D line segment connecting two endpoints P_0^i and P_1^i , which is used to refine the positions of vertices of its corresponding mesh line F_i (Fig. 2b).

With such a surface scaffold structure constructed from input mesh and images, we achieve the model regularization by forcing the mesh lines to be straight, i.e., projecting their mesh vertices to their corresponding scaffold lines, and smoothing the remaining mesh parts.

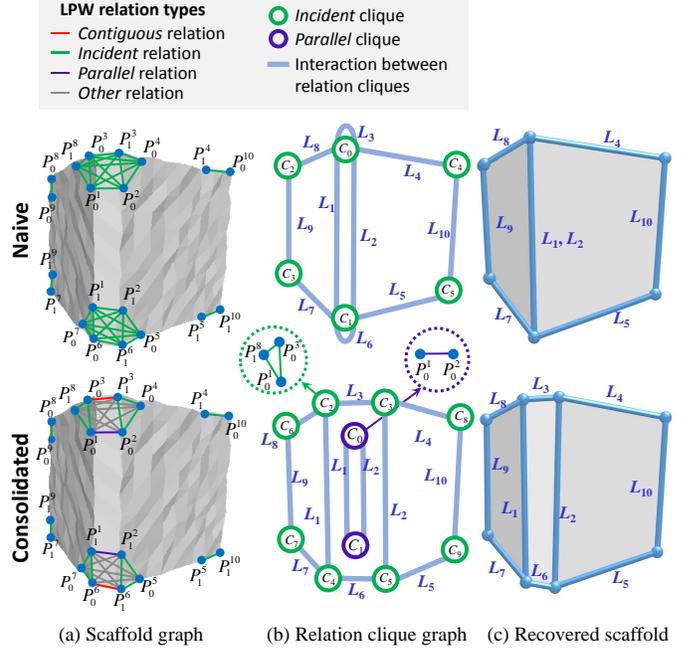


Fig. 3. Scaffold graph representation. A scaffold graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ represents and optimizes the scaffold topology, where \mathcal{V}_s contains all endpoints of scaffold lines and \mathcal{E}_s contains all local point-wise (LPW) relations. The graph is first constructed by connecting two endpoints whose distance is less than ϵ_d (Fig. 2b), and then is added with complemented edges such that each connected component forms a complete subgraph. (a) Scaffold graphs \mathcal{G}_s with all LPW relation types labeled overlaid on the input 3D model. (b) Relation clique graphs \mathcal{G}_c , where each relation clique is a complete subgraph of the same relation type. (c) Recovered scaffolds \mathcal{S} overlaid on the resulting 3D models according to the labeled relation types in (a). In the naive scaffold graph, all the LPW relations are simply assigned as *incident*, leading to an unfaithfully recovered scaffold (L_3 and L_6 are missing). In the consolidated scaffold graph, the relation types are labeled correctly and the recovered scaffold presents a faithful shape. (Best viewed in color.)

2.2 Scaffold Topology

Since scaffold lines are obtained by back-projecting 2D lines onto a rough mesh, their poses (positions and orientations) are erroneous due to the mesh noise and registration errors, presenting disordered structures (Fig. 2b). Intuitively, the regularity can be substantially enhanced by imposing the scaffold lines to satisfy the correct topology, such as forcing incident lines to intersect at a common point and parallel lines to be aligned in a shared orientation. Thus we are motivated to first optimize the scaffold topology.

Often, 3D lines are determined to be incident lines if their endpoints are close enough. However, because of the erroneous positions of the endpoints, some close parallel lines can be mis-determined to be incident lines. Thus, the proximity of endpoints is insufficient for determining whether lines are incident or not. Therefore, we need to accurately determine the types of spatial relations between the nearby endpoints, so as to optimize the scaffold topology.

Local point-wise (LPW) relation. Considering the position proximity prerequisite, we introduce an LPW relation $r = (P_z^i, P_z^j)$ between two endpoints P_z^i and P_z^j whose distance is less than a threshold ϵ_d (orange arrows in Fig. 2b). LPW relations are defined as being of the following four types (see examples in Fig. 2b):

- *Contiguous*. Two endpoints belong to the same line, e.g., P_0^3 and P_1^3 .
- *Incident*. Two endpoints belong to different lines but have the same position, e.g., P_0^1 and P_0^3 .
- *Parallel*. The closest pair of endpoints of two parallel (but non-collinear) lines, e.g., P_0^2 and P_0^1 . P_0^1 and P_1^2 are not *parallel* because they are not the closest pair.
- *Other*. Not any of the above three types, e.g., P_0^1 and P_1^3 .

Scaffold graph. A scaffold graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ is used to represent and optimize the scaffold topology, where the vertex set contains all endpoints of the scaffold lines $\mathcal{V}_s = \{P_z^i | z \in \{0, 1\}, i \in [1, N]\}$ and the edge set is initially constructed as $\mathcal{E}_s = \{r = (P_z^i, P_{z'}^j) | dist(P_z^i, P_{z'}^j) < \epsilon_d\}$ which contains all the LPW relations. At the beginning, the types of LPW relations are unknown, and our goal is to correctly derive them to optimize the scaffold topology.

Relation clique. Owing to the transitivity of incidence, any two endpoints that are connected by an *incident* path should be *incident*, e.g., P_0^1 and P_1^8 in Fig. 2b. Thus, a connected component connected by *incident* relations in \mathcal{G}_s should form a clique. This is the same as *parallel* relations. We call such *incident* and *parallel* cliques *relation cliques*.

We observe that the original edge set \mathcal{E}_s does not completely capture all the relations, some of which are produced by the transitivity, e.g., P_0^1 and P_1^8 . So as not to miss any potential relations, we add complementary edges into \mathcal{E}_s to make each connected component a complete subgraph, so that any relation between two endpoints that are connected with a path can be captured. Scaffold graph examples with complementary edges are shown in Fig. 3a, and the types of the relations are naively labeled and finally consolidated respectively.

Relation clique graph. Given a scaffold graph with labeled relation types, each potential *incident* clique is found by extracting any *incident* connected components, similarly as a *parallel* clique. However, the types of relations may not all be labeled correctly. For example, an edge in an *incident* connected component may be labeled as *parallel*, thus the edge is also in a *parallel* clique. Besides, two labeled relations connecting two pairs of endpoints that respectively belong to two scaffold lines may lead to a contradiction (e.g., P_0^i and P_0^j are *incident*, but P_1^i and P_0^j are *parallel*) or a non-existent (e.g., P_0^i and P_0^j are *incident*, and P_1^i and P_0^j are also *incident*). These invalid situations, termed topological conflicts, can be represented by the interactions at the clique level. We use a relation clique graph $\mathcal{G}_c = (\mathcal{C}_I \cup \mathcal{C}_P, \mathcal{E}_c)$ to encode and resolve such topological conflicts, where \mathcal{C}_I and \mathcal{C}_P denote the extracted *incident* clique set and *parallel* clique set respectively. In \mathcal{G}_c , two cliques are connected by an edge, if there is a relation both in them or a scaffold line linking them (e.g., L_1 links C_0 and C_1 in bottom of Fig. 3b). Consequently, self edges of a clique or multiple edges between two cliques may exist in \mathcal{G}_c . Fig. 3b shows the clique graph deduced from the corresponding labeled scaffold graph in Fig. 3a.

In Fig. 3, the scaffolds (c) are recovered from the scaffold graphs with all relations labeled (a). In the naively labeled scaffold graph (top), the relations between P_0^1 and P_0^2 , P_0^1 and P_1^2 are simply assigned as *incident*, leading to L_3, L_6 dis-

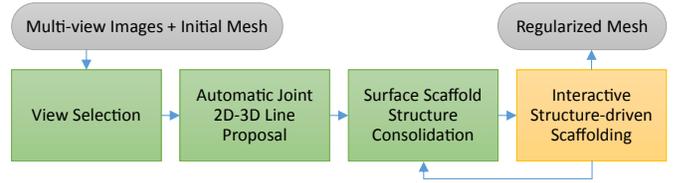


Fig. 4. Workflow of the proposed approach. The automatic phase is colored green and the interactive phase is colored yellow.

appearing in the recovered scaffold. If the scaffold topology is optimized, i.e., the relation types are derived correctly, the recovered scaffold will present the correct shape (bottom).

2.3 Workflow

The workflow of our semi-automatic system is illustrated in Fig. 4, and an example from our dataset is shown in Fig. 1. Our system takes a set of images with calibrated cameras and an initial mesh reconstructed by any stereo-based method as the input (we use [1]). At the preprocessing step, we fill small holes detected by the work [28] with flat surfaces.

Automatic phase (green in Fig. 4). A minimum set of views with accurate camera geometry is first selected by simplifying a view redundancy graph, following the work in [29]. Second, we automatically propose as many scaffold lines as possible from the selected views using the photometric and geometric information jointly to form an initial scaffold line set. Such scaffold lines are used to construct an initial scaffold graph where the types of LPW relations are currently unknown. The scaffold topology is then optimized by deriving the all the types of LPW relations using a non-linear programming method for best fitting the input mesh. To satisfy the optimized topology, the poses of the scaffold lines are refined in a least-square manner, and the corresponding mesh lines are adjusted accordingly (Fig. 1b) by projecting their contained vertices onto the refined scaffold lines.

Interactive phase (yellow in Fig. 4). The scaffold line set grows incrementally with the help of the user intervention. Guided by the scaffold structure, scaffold lines symmetric to the user input lines are automatically proposed using a graph matching method. By iteratively performing the interactive stage and the scaffold consolidation, the scaffold become complete and the mesh evolves into a regular one (Fig. 1d).

Finally, the regularized model is textured (Fig. 1e) and further applications such as simplification (Fig. 1f) and non-photorealistic rendering can be significantly improved based on the regularization.

3 AUTOMATIC JOINT 2D-3D LINE PROPOSAL

Extracting linear features \mathcal{F} directly from the input mesh relies strictly on the mesh quality, while the reconstructed meshes are inevitably noisy and incomplete. We leverage the 2D line detection from images of sufficient quality and resolution, and then obtain 3D linear features by back-projecting the 2D lines to the 3D surface. However, image lines are extracted with texture discontinuities or shading

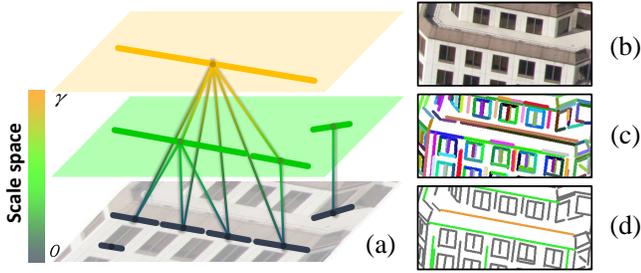


Fig. 5. Accurate and coherent (not fragmented) 2D line extraction at multi-scales. (a) Line extraction from coarse to fine image scales (from top to bottom). The thin edges connect collinear and overlapped lines. Each connected group of the lines tends to describe a single line. Here shows three such groups. (b) Input image. (c) LSD [31] result at finest scale, extracting over-fragmented and even spurious lines (close-up of Fig. 6b). (d) Accurate and coherent 2D line extraction using our method, exhibiting cleaner result and preserving high line extraction accuracy (close-up of Fig. 6c). The line colors indicate the image scale space in (a).

variances, which do not all represent structures. Some 2D lines may even erroneously correspond with 3D lines due to the registration error. In order to propose the desired scaffold lines, not only do we start to extract them from images, but also verify them using 2D photometric and 3D geometric information jointly.

First, to improve the efficiency in the subsequent scaffold line proposal, we automatically select a minimum set of views. Then, from the selected views, we extract high-quality 2D lines as the candidates for the proposal. Finally, a binary classifier is constructed to filter out the candidates that do not contribute to the scaffold structure using the AdaBoost algorithm [30].

3.1 View Selection

Obviously, the input views which can reconstruct the whole surface contains a lot of redundancy for only proposing a few structural linear features. To improve the proposal efficiency, it is necessary to first select a small set of appropriate views.

Following previous work [29] which selects views for reducing multi-view reconstruction redundancy, we similarly perform the view selection by simplifying a view redundancy graph. Differently, we select a set of views that are 1) most accurately calibrated, and 2) with the minimum size to satisfy the completeness condition, i.e., covering the mesh regions where all structural linear features are potentially located. Such potential regions which cover linear features are discovered to be non-planar regions. The planarity is evaluated using the method in [3]. This graph simplification problem can be approximately solved by greedily deleting the most inaccurately calibrated view when the completeness condition is not violated.

3.2 Accurate and Coherent 2D Line Extraction

From the selected views, we then aim to extract 2D line candidates for proposing the 3D linear features that can represent the building structure concisely and completely. Traditional line detectors tend to generate fragmented lines at fine image scales and inaccurate ones at coarse scales.

In addition, detection at the object level [33] incorporating complex features and classifiers is not appropriate for our case due to the high computation complexity.

An intuitive way to efficiently extract both an accurate and coherent (not fragmented) line is to find the best image scale from where the desired line can be extracted. First, we extract 2D lines in an image pyramid with the scale factors $\Omega = \{\theta^0, \theta^1, \dots, \theta^\gamma\}$ using [31] ($\theta = 0.8$ in our implementation). It is observed that the detected multi-scale lines that are collinear and overlap tend to describe a single line (see examples in Fig. 5a). We then group such collinear lines, and extract each final line from the image region covered by a group of lines. In such an image region σ , we evaluate the lines from each scale $s \in [0, \gamma]$, and find the best scale by maximizing the association of the following two terms: 1) an accuracy term $A(s)$ measuring the directional edge strength along the directions of the extracted lines, and 2) a coherence term $U(s)$ measuring the degree of fragment, i.e., the entropy of the extracted lines, at scale s .

$$A(s) = -\log\left(\frac{1}{|\sigma|} \sum_{l \in \mathcal{L}_\sigma(s)} \sum_{x \in l} \nabla_{\mathbf{v}_l} I(x)\right) \quad (1)$$

$$U(s) = -\sum_{l \in \mathcal{L}_\sigma(s)} P(l) \log P(l) \quad (2)$$

where $\mathcal{L}_\sigma(s)$ is a subset of the grouped lines in region σ at scale s , \mathbf{v}_l the directional vector of line l , $\nabla_{\mathbf{v}_l} I(x)$ the directional derivatives at pixel x of image I , $|\sigma|$ the number of covered pixels by σ , $P(l)$ the probability of the line l located in σ , computed as the ratio of the occupied area.

A sample of extracted 2D lines is shown in Fig. 6c (close-up in Fig. 5d). Compared with the traditional edge extraction methods, Canny [32] (Fig. 6a) and LSD [31] (Fig. 6b, close-up in Fig. 5c), our result is much cleaner (cluttered lines are removed) and more coherent (long lines are not fragmented) while preserving the extraction accuracy at fine scales. These extracted 2D lines are used as candidates for proposing scaffold lines.

3.3 Learning-based Line Filtering

The extracted 2D lines include a lot of textural edges which do not represent the building structure and occluding contours which often have vulnerable 2D-to-3D correspondences and may lead to erroneous back-projecting results. Therefore, filtering the extracted 2D lines before constructing the scaffold is necessary to ensure that the proposed scaffold lines are structural and reliable.

With the input mesh, we can eliminate textural edges according to the geometric saliency. Besides, the accuracy of 2D-to-3D correspondences can be evaluated with the multi-view photo consistency [6]. A binary classifier is constructed for identifying whether the lines should be filtered out, using the AdaBoost algorithm [30], which can well combine various weak classifiers. Table 1 lists the considered weak classifiers and features. The 3D geometry features are extracted from the mesh vertices which are covered by back-projecting the 2D line with the constant width (7 pixels for a 5616 * 3744 image) onto the mesh.

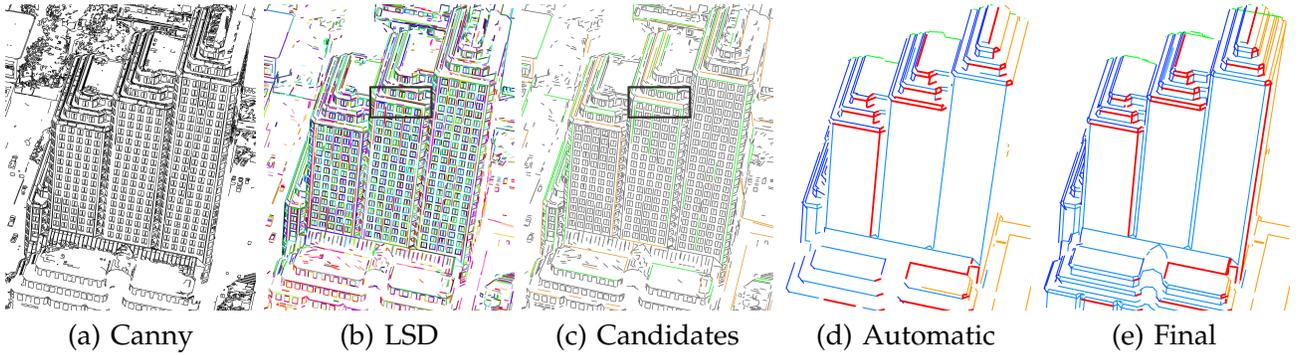


Fig. 6. Extraction of structural 2D lines. (a) Canny edge detection [32]. (b) LSD [31]. (c) Extracted accurate and coherent 2D lines serving as candidates for the scaffold line proposal. The line colors are related to the scale space described in Fig. 5a. (d) Projection of the proposed scaffold lines after the automatic phase. 2D lines of the same color are the projections of the scaffold lines proposed from the same view. Red lines are from the current view. Our learning-based filtering is able to eliminate the textural lines and preserve the multi-view consistency, i.e., the projections of the scaffold lines proposed from other views (non-red) are almost accurate at the positions in the current view. (e) Projection of the final scaffold, presenting the global structure of the building precisely and concisely.

TABLE 1

Weak classifiers for filtering unexpected lines. “Hist” is short for “histogram”. “Planarity” and “PCA” represent the planarity measure and principal component analysis of the covered mesh vertices respectively.

Weak classifier	Feature	Dimension
Geometric saliency	Planarity [3] Hist.	11×3
Spatial linearity	Singular values of PCA	3
MV consistency	Reprojection error [6] Hist.	11

Fig. 6 shows the comparison of extracting structural 2D lines using different methods. We find that our method is able to filter out the textural lines and preserve multi-view consistency of the proposed lines. In (d), lines automatically proposed from other views (non-red) are also projected onto the current view. Their projections are coincidental with edges in the current view, showing that our method ensures high multi-view consistency.

4 SURFACE SCAFFOLD STRUCTURE CONSOLIDATION

After the image lines are extracted, we first generate a set of mesh lines and initialize the corresponding scaffold lines by back-projecting the image lines onto the input mesh. With the initial scaffold lines, we then optimize the scaffold topology by deriving the types of LPW relations that best fit the input mesh, using nonlinear programming. Finally, the global shape of the scaffold structure is adjusted to satisfy the optimized topology in a least-square manner.

4.1 Mesh Line Generation and Scaffold Line Initialization

A mesh line F which is a sequence of mesh edges represents a 3D linear feature, and its corresponding scaffold line L is used to refine the positions of the vertices of F . First, we back-project an image line l with the constant width (same in Section 3.3) onto the initial mesh \mathcal{M} , which results in a set of covered mesh vertices \mathcal{P}_l , as illustrated in Fig. 7a. The scaffold line L is then initialized by fitting L to \mathcal{P}_l . Finally, the mesh line F is extracted as a forward path closest to L , i.e., the total distance from its vertices to L is the smallest.

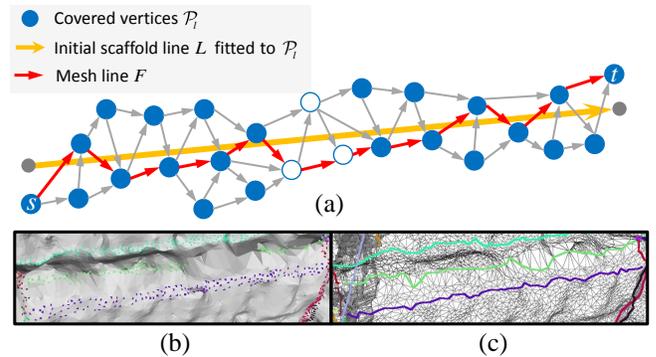


Fig. 7. Mesh line generation and scaffold line initialization. (a) We back-project an image line l onto the mesh \mathcal{M} and get a set of covered vertices \mathcal{P}_l . The scaffold line L is fitted to \mathcal{P}_l . The mesh line F is extracted as a forward path closest to L . (b) Covered mesh vertices by back-projecting image lines onto the mesh. (c) Extracted mesh lines on the irregular mesh using Algorithm 1.

We construct a weighted directed graph \mathcal{G}_l based on the input mesh topology to extract F . The details are described in Algorithm 1.

4.2 Scaffold Topology Optimization

The initial scaffold lines are in inaccurate poses because of the mesh noise and registration errors (e.g., Fig. 2b). In order to refine the initial scaffold, we aim to first find the correct scaffold topology, and then adjust the scaffold shape accordingly. As mentioned in Section 2.2, the types of LPW relations in the scaffold graph interpret the scaffold topology. Let $\mathbf{R}_s = [R_1, \dots, R_k, \dots, R_{N_s}]$, $N_s = |\mathcal{E}_s|$, denote an LPW relation labeling over the scaffold graph edge set \mathcal{E}_s . The goal of scaffold topology optimization is to associate each edge (LPW relation) $r_k \in \mathcal{E}_s$ with a label $R_k \in \{y_C, y_I, y_P, y_O\}$ respecting the input mesh, denoting *contiguous*, *incident*, *parallel* and *other* relations respectively.

However, labeling each R_k independently, which only considers the local geometry in the region related to r_k , is sensitive to mesh noise. Since the LPW relation labels are conditionally dependent on others due to the transitivity of incidence/parallelism and the contiguity of endpoints

Algorithm 1 Mesh Line Generation and Scaffold Line Initialization

- 1: **Input:** 2D line l in view I and rough mesh $\mathcal{M} = \{\mathcal{V}_m, \mathcal{E}_m\}$.
Output: Mesh line F and scaffold line L .
- 2: Back-project l onto \mathcal{M} , obtaining the covered mesh vertex set \mathcal{P}_l (Fig. 7b).
- 3: Fit the initial scaffold line L to the mesh vertices in \mathcal{P}_l using RANSAC [34].
- 4: Construct a weighted directed graph from \mathcal{M} , $\mathcal{G}_l = (\mathcal{V}_m, \mathcal{E}_l)$.

$$\mathcal{E}_l = \{e = (a, b) | e \in \mathcal{E}_m \text{ and } \mathbf{v}_L \cdot \mathbf{v}_e \geq 0\}$$

$$w_e = \text{dist}(b, L)$$

where \mathbf{v}_L denotes the direction of L .

- 5: Denote s, t as the nearest mesh vertices to the two endpoints of L respectively. Extract the shortest path from s to t over the graph \mathcal{G}_l as the mesh line F . (Fig. 7c)
-

(linked by the same scaffold line), large noise that introduces a lot of labeling errors can lead to many topological conflicts.

We transform the scaffold topology optimization problem, which aims to maximize the building regularity, to a constrained energy minimization problem where the constraints are incorporated to avoid topological conflicts. We then iteratively correct the most likely mis-determined labels which cause the topological conflicts so as to solve the minimization.

4.2.1 Energy Function Formulation

As discussed above, independent labeling may lead to topological conflicts. Therefore, the scaffold topology optimization should simultaneously take data fidelity and topological constraints into account. This problem can be formulated as a nonlinear programming problem which involves minimizing the labeling cost with respect to the input mesh while not introducing any topological conflicts. Nonlinear programming takes the form:

$$\begin{aligned} \mathbf{R}_s^* &:= \arg \min_{\mathbf{R}_s} \sum_{R_k \in \mathbf{R}_s} \psi(R_k) \\ \text{s.t. } &\Phi(C_i, C_j) = 0, \forall C_i \bowtie C_j \end{aligned} \quad (3)$$

where $\psi(R_k)$ denotes the individual cost of the label R_k at relation r_k , $C_i \bowtie C_j$ the adjacent relation clique pairwise set, $\Phi(C_i, C_j)$ the penalty of the topological constraint on the adjacent relation cliques. Note that the *contiguous* relation is predetermined because it is certain whether two endpoints belong to a scaffold line. We then only need to label the other three relation types.

Topological Constraints. As described in Section 2.2, the topological constraints are modeled on the interactions between adjacent relation cliques in the clique graph \mathcal{G}_c . An edge connecting two cliques means that a relation is both in the two cliques or a scaffold line links the two cliques. This is because such edges can capture and cover all topological conflicts, namely, 1) a contiguous relation is deduced to be *incident* by the transitivity, which results in a self-edge connecting an *incident* clique (Fig. 8a); 2) two

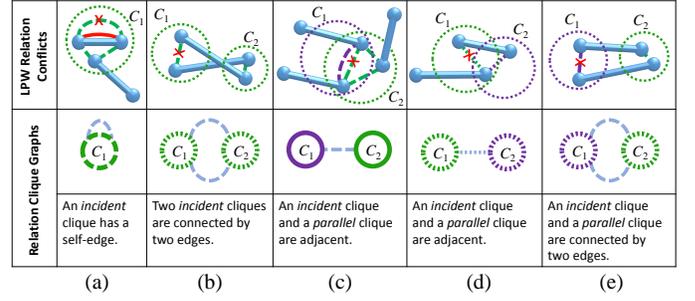


Fig. 8. Enumeration of the topological conflicts and their relation clique graphs used for checking the conflicts. Symbols are defined the same as in Fig. 3. The dashed circles in the first row bound the endpoints that belong to the relation cliques. The solid edges are the labeled relations, and the dashed edges are the deduced labels from the relation transitivity. The red crosses denote the wrong labels that cause the conflicts.

endpoints of a line are *incident* with other two endpoints of another line respectively, leading to two different edges connecting the same pair of two *incident* cliques (Fig. 8b); 3) a relation is labeled *incident* but deduced *parallel* by the transitivity, or vice versa, consequently, this relation exists both in an *incident* clique and a *parallel* clique, leading to the two cliques being adjacent (Fig. 8c, d, e).

By checking the interactions between all adjacent cliques in \mathcal{G}_c , we can discover all the topological conflicts according to the enumerated description in Fig. 8. Then, the penalty $\Phi(C_i, C_j)$ is assigned 1 if there are topological conflicts between C_i and C_j , otherwise $\Phi(C_i, C_j) = 0$.

Data cost. The individual data cost $\psi(R_k) = -\log(P(R_k|r_k))$ evaluates the fit of the label R_k to the local data observation related to r_k , where $P(R_k|r_k)$ is the probability of labeling r_k as R_k .

For the *parallel* relation, we should not only consider the angle between the two corresponding lines, but also should exclude the collinear case which is considered to be an *incident* relation. Inspired by the discussion on 1D line-line topological relations in [35], we find that in two parallel 3D lines, the closest pair of the endpoints favors head-to-head or tail-to-tail alignments. Thus, we project two 3D lines of the same orientation to an appropriate \mathbb{R}^1 space and check the alignments of their endpoints so as to infer whether the two lines are more likely to be collinear or parallel.

Considering the two possible parallel lines L_i and L_j , we use their overlapping ratio of the projections on a reference line, $\rho(r_k) \in [0, 1]$, to measure the confidence of being the favored alignments, namely $\rho(r_k) = \frac{|L_i^* \cap L_j^*|}{|L_i^* \cup L_j^*|}$, where L_i^* and L_j^* denote the projections of L_i and L_j on the reference line L^* (L^* is fitted to all the vertices of the two corresponding mesh lines F_i and F_j).

Then, the probability of assigning R_k as *parallel* is defined as

$$P(R_k = y_P|r_k) = [\rho(r_k)]^\alpha \cos \theta \quad (4)$$

where θ denotes the angle between L_i and L_j , and α the parameter controlling the impact of the overlapping ratio (1/3 in our implementation).

Two endpoints connected by r_k with position proximity (their distance d_{r_k} is smaller than ϵ_d) tend to be *incident*

except when they belong to the same line or two parallel lines. Thus the probability of being *other* relation will be close to 0 when $d_{r_k} < \epsilon_d$. On the contrary, it should be a large value and increase with the distance d_{r_k} . Note that d_{r_k} may be greater than ϵ_d because some complementary relations are added in \mathcal{E}_s as described in Section 2.2. Intuitively, we use the distance transformation criteria [36] to define the probability of assigning R_k as *other*:

$$P(R_k = y_O | r_k) = 1 - \beta e^{\min(0, \epsilon_d - d_{r_k})} \quad (5)$$

where β denotes the balancing parameter (0.96 in our implementation).

Since *contiguous* relations are predetermined, for a relation r_k that does not connect two endpoints of a scaffold line, the probability of labeling R_k as *incident* is $P(R_k = y_I | r_k) = 1 - P(R_k = y_P | r_k) - P(R_k = y_O | r_k)$.

4.2.2 Energy Minimization

Owing to neither the objective function nor the constraint set in (3) being convex, no polynomial-time methods for solving this non-linear programming problem exist. Instead, we propose an approximation algorithm which is inspired by the gradient descent methods. But rather than taking steps proportional to the negative of the gradient, we apply the min-cut algorithm to resolve the conflicts by correcting the most likely mis-determined labels iteratively.

In each iteration, we first find the most likely mis-determined relation $r_c = (s_c, t_c)$ with label $R_c = y_c$ that causes the topological conflict according to Fig. 8, and then correct it. However, if there are some paths with all edges labeled y_c from s_c to t_c , the newly updated label R_c may introduce new conflicts due to the relation transitivity (we consider *incident* and *parallel* here). Thus, we also change the labels of edges in a cut from s_c to t_c to prevent such paths from being connected. We select the min-cut with the edge weight $P(R_k = y_c | r_k)$ for changing the labels, so that the labels with large data costs tend to be changed.

Furthermore, in order to improve the efficiency of the minimization, we speed it up by reducing the size of the objective LPW relation set \mathcal{E}_s . Relation transitivity may not only lead to topological conflicts, but also produce a large relation redundancy. Finding a minimum LPW relation set \mathcal{E}_o which can derive all the other labels of the relations in \mathcal{E}_s is attractive for speeding up the algorithm. In a connected component in \mathcal{G}_s , a spanning tree with all relations labeled *incident* decides that all the other relations are *incident*. Thus, for each component, we extract a candidate set to be labeled with a Minimum Spanning Tree (MST) using the edge weight $-\log(P(R_k = y_I | r_k))$. This is because the MST contains the edges that are most likely to be labeled *incident*, and then we can reduce the iterations of updating the labels to be *incident*. Similarly, for *parallel* relations, the corresponding MST can be extracted. Therefore, the objective relation set \mathcal{E}_o can be constructed as the union of the relations in all *incident* and *parallel* MSTs. Then, the objective function of (3) is equivalent to

$$\mathbf{R}_o^* := \arg \min_{\mathbf{R}_o} \sum_{R_k \in \mathbf{R}_s} \psi(R_k) \quad (6)$$

Algorithm 2 Scaffold Topology Optimization

Require: Objective edge set \mathcal{E}_o and their label set $\{y_C, y_I, y_P, y_O\}$.

Ensure: The target labeling \mathbf{R}_o^* of Equation 3.

- 1: **Initialization.** $\mathbf{R}_o^* = \arg \min_{\mathbf{R}_o} \sum_{R_k \in \mathbf{R}_o} \psi(R_k)$ without any constraints.
Initialize a priority queue $Q_k = [y_1, y_2, y_3]$, $y_j \in \{y_I, y_P, y_O\}$ for each $r_k \in \mathcal{E}_o$ with the decreasing order of $P(R_k = y_j | r_k)$, $j = 1, 2, 3$.
- 2: Deduce \mathcal{C}_I and \mathcal{C}_P from \mathcal{G}_s according to \mathbf{R}_o^* , and construct the clique graph \mathcal{G}_c .
- 3: Deduce labels \mathbf{R}_s of the relations in \mathcal{E}_s from \mathbf{R}_o^* .
- 4: **while** one topological conflict is found according to \mathcal{G}_c
do
- 5: Find the edge $r_c = (s_c, t_c) \in \mathcal{E}_s$ with the minimum labeling probability $P(R_c = y_c | r_c)$ among the edges that cause the conflict. (red cross in Fig. 8)
- 6: Extract a min-cut \mathcal{E}_{cut} from s_c to t_c in a weighted graph $(\mathcal{V}_s, \mathcal{E}_o)$ with the weight set $\{P(R_k = y_c | r_k) | r_k \in \mathcal{E}_o\}$.
- 7: Delete the queue head $head(Q_k)$ for each relation $r_k \in \mathcal{E}_{cut}$, and update $R_k = head(Q_k)$.
- 8: **end while**

where $\mathbf{R}_o = [R_{o_1}, \dots, R_{o_k}, \dots, R_{o_N}]$, $r_{o_k} \in \mathcal{E}_o$ denotes a relation labeling of the objective relations set \mathcal{E}_o . \mathbf{R}_s is deduced from \mathbf{R}_o .

Algorithm 2 details the minimization process.

4.3 Global Shape Refinement

Once the scaffold topology is optimized, the global shape of the mesh \mathcal{M} can be refined by adjusting the positions and orientations of the scaffold lines to satisfy the optimized topology and applying feature-preserving smoothing.

To respect the optimized topology, the position of each endpoint should be moved to an incident point c_i of a related *incident* cliques $C_i \in \mathcal{C}_I$. The orientation of each scaffold line \mathbf{v}_L should be aligned to a shared orientation \mathbf{v}_j with all the lines belonging to a *parallel* clique $C_j \in \mathcal{C}_P$. (For generality, we associate an endpoint that is not *incident* with any others with a virtual *incident* clique that contains only one element, similar to the *parallel* cliques.) Then, each scaffold line can be represented as $c_i + \lambda \mathbf{v}_j$. Our goal is to find the best incident points $\{c_i\}$ of all *incident* cliques in \mathcal{C}_I and the directions $\{\mathbf{v}_j\}$ of all *parallel* cliques in \mathcal{C}_P . The formulation is to minimize the association of all the distances between the scaffold lines and their incident points and all the differences between the directions of the scaffold lines and their shared directions:

$$\min_{\{c_i\}, \{\mathbf{v}_j\}} \left(\sum_{C_i \in \mathcal{C}_I} \sum_{L \in C_i} \omega(L) \text{dist}(c_i, L) + \sum_{C_j \in \mathcal{C}_P} \sum_{L \in C_j} \omega(L) \|\mathbf{v}_j - \mathbf{v}_L\|_2 \right) \quad (7)$$

where weight $\omega(L)$ denotes the quality of a proposed scaffold line L , which is implemented as the least-square fitting confidence of L to \mathcal{P}_l (computed in Section 4.1).

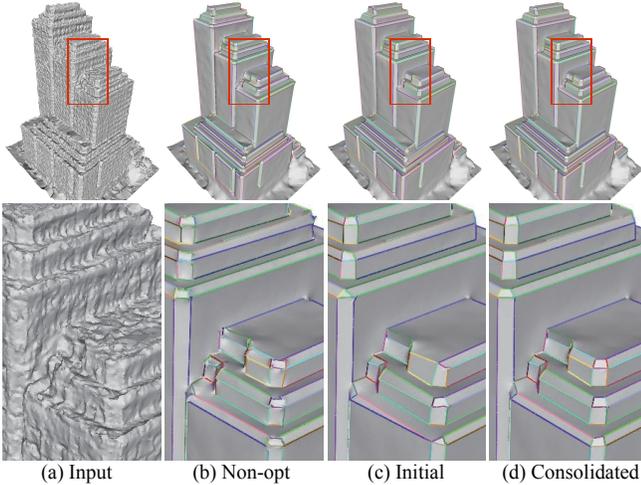


Fig. 9. Scaffold topology optimization and global shape regularization results. (a) Input mesh. (b) Regularized mesh based on the scaffold lines without topology optimization. (c) Regularized mesh based on the scaffold with initial topology (line 1 in Algorithm 2). (d) Regularized mesh based on the optimized scaffold.

We solve (7) by iteratively computing the optimal directions $\{\mathbf{v}_j\}$ of the *parallel* cliques when fixing the positions of the incident points $\{c_i\}$ using the least-square method, and then computing the optimal $\{c_i\}$ when fixing $\{\mathbf{v}_j\}$. The iterative process stops when the change of Equation (7) is less than 10^{-4} (often converges after two or three iterations in our experiments). The convergence threshold is generally used across different datasets. Note that the model is normalized by its bounding box.

After the incident points and parallel directions in the relation cliques are computed, the mesh line F_i corresponding to each newly computed scaffold line L_i is updated by projecting each vertices of F_i to L_i . Finally, other parts of the mesh are smoothed using the Least-squares Meshes [37] method to highlight the regular structures.

The result of the surface scaffold structure consolidation is shown in Fig. 9. Compared with the result without topology optimization (b) and the result with the initial scaffold (c), our result (d) better fits the data.

5 INTERACTIVE STRUCTURE-DRIVEN SCAFFOLDING

After the automatic phase, the scaffold \mathcal{S} is approaching a complete one, i.e., the scaffold line set \mathcal{L}_s containing most of the desired lines (Fig. 1b). Owing to automatic methods often being influenced by image quality and variances in edge strength on complex scenes, having a few lines missing is inevitable. To make the scaffold more complete, we exploit user intelligence and provide a well-designed user interface for iteratively adding the missing scaffold lines. In each iteration, the user draws a 2D line in a selected view, then our system automatically adds its corresponding scaffold line into \mathcal{L}_s and consolidate the new \mathcal{S} . Furthermore, to minimize user intervention in adding missing lines, we utilize symmetry, one of the most prevalent features of urban scenes, to improve the efficiency in the line proposal. Given the user input line, our system proposes more lines

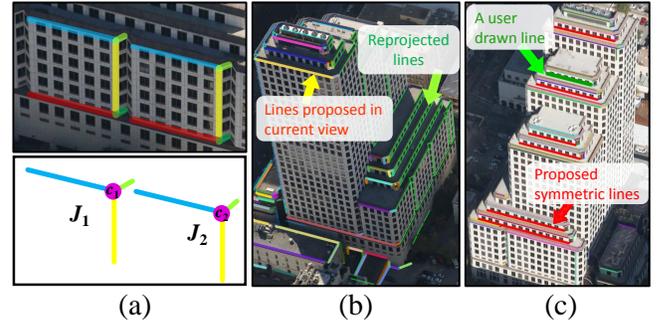


Fig. 10. Interactive scaffold line proposal. (a) Junction structures J_1 and J_2 are isomorphic. c_1 and c_2 are junctions of J_1 and J_2 respectively. Lines of the same color are symmetric. (b) 2D UI illustration. The projection of the scaffold overlaid on the image shows overall perspective of the scaffold in the current view and indicates users where lines are absent. Thick lines in random colors are proposed from the current view. Thin lines are the reprojected scaffold lines proposed from other views. Colors of the reprojected lines denote the views which they are proposed from. (c) An iteration of interactive structure-driven line proposal.

that are symmetric (e.g., the two yellow lines in Fig. 10a are symmetric) to the user input by connecting potential pairs of incident points in \mathcal{S} . Fig. 10c illustrates an interactive iteration.

5.1 User Interface

Our interactive system integrates 2D and 3D interface for both visualization and editing. The 2D interface (Fig. 1c and Fig. 10b) can assist the user to visualize, create and edit 2D lines, which correspond to the scaffold lines. All the proposed scaffold lines are projected onto the current view to give the user an overall perspective of the scaffold shape, so that the user can identify where any lines are missing and add them in. Edge snapping [12] and curve segmentation [38] are also integrated into the user interface for interaction efficiency and convenience. The 3D interface shows the current consolidated scaffold and regularized model for user’s reference (Fig. 1d). Please see the supplementary material for more details.

5.2 Symmetry-based Scaffold Line Proposal

Symmetries, with strong coherence both in texture and structure, are prevalent on urban scenes. Thus, we exploit this strong urban prior to minimize the user efforts of proposing missing scaffold lines. When the user draws an exemplar line \bar{l} , its corresponding scaffold line \bar{L} is generated and consolidated in the scaffold. Then, our system automatically proposes symmetric lines that have strong similarities to \bar{L} .

Scaffold line similarity. We embrace the observation that symmetric scaffold lines exhibit similar textural appearances in the images and isomorphic structures (see the two yellow lines in top of Fig. 10a). Currently, the scaffold structure is consolidated, which can tell the spatial relation types between the nearby scaffold lines. The similarity between two scaffold lines in a view is characterized by the textural similarity of both themselves and other lines that they are incident with. We use a *juncture structure* J to assemble a set of incident scaffold lines $\mathcal{L}(J)$ connected by an incident

point c (bottom of Fig. 10a). Each scaffold line L has exactly two related junction structures, J_A and J_B . The scaffold line similarity between L and L' is defined regarding to both the textural similarity of the two junction structure pairs $\delta(J_A, J'_A)$ and $\delta(J_B, J'_B)$, and that of the individual lines $\delta(L, L')$.

The textual similarity of the two individual scaffold lines $\delta(L, L')$ is defined as the dot product of MSLD descriptors $f(\cdot)$ [39] of the two corresponding image lines, $\delta(L, L') = f(l) \cdot f(l')$ if $f(l) \cdot f(l') > 0.5$, otherwise $\delta(L, L') = 0$ (low similarity is omitted).

Junction structure similarity. The junction structure similarity $\delta(J, J')$ is the sum of the textural similarity of all the matched line pairs in $\mathcal{L}(J)$ and $\mathcal{L}(J')$

$$\delta(J, J') = \max_m \sum_{L \in \mathcal{L}(J)} \delta(L, m(L)) \quad (8)$$

where m is the mapping function $m : \mathcal{L}(J) \mapsto \mathcal{L}(J')$. The optimal mapping which best matches the lines in the two sets $\mathcal{L}(J)$ and $\mathcal{L}(J')$ can be found by solving the maximum weighted matching problem [40].

Scaffold line proposal. The search space of symmetric lines in either the images or the mesh is huge. We find that the existing scaffold offers most of the incident points, which can serve as the endpoints of the potential scaffold lines. Given the consolidated scaffold \mathcal{S} , we first extract a set of junction structures \mathcal{J} that contains all the junction structures in \mathcal{S} and its corresponding incident point set \mathcal{P}_c . Then we construct a set of candidate scaffold lines $\mathcal{L}_{can} = \{L = (c_A, c_B) | c_A, c_B \in \mathcal{P}_c\}$ by connecting every two incident points c_A, c_B (with the corresponding junction structures J_A and J_B). Note that only visible scaffold lines in the current view are considered. A new scaffold line L will be proposed from \mathcal{L}_{can} if it satisfies the following two conditions, 1) its two junction structure pairs are highly similar to the exemplar ones, i.e., $\delta(J_A, \bar{J}_A)$ and $\delta(J_B, \bar{J}_B)$ are both greater than a threshold (1.1 in our implementation); 2) its individual texture is highly similar to the exemplar one, $\delta(\bar{L}, L) > 0.8$.

Since the symmetric lines are proposed based on the existing scaffold, there are very few mis-proposed lines. In case of proposing unexpected lines, the user can manually delete them. But the deletion operations are of a very small portion of the user efforts, shown in the experiments (Table 2).

After each interaction step, the scaffold is updated by adding newly proposed lines and reperforming the consolidation (Section 4). The consolidation process including solving the nonlinear programming (3) and the least-square optimization (7) is very efficient (taking 0.015s in average) due to the small relation set (smaller than 1000), and thus the response time is in realtime. Fig. 10c illustrates an iteration of the interactive structure-driven line proposal. Iteratively, the surface scaffold structure evolves by performing the interactive structure-driven scaffolding and the scaffold consolidation.

6 EXPERIMENTS

We implement the system with C++ and test it on a PC with Intel Core-i3 3.10GHz and 16GB RAM. Quasi-dense structure from motion [1] is used to estimate the poses

of the input images and the dense points. Poisson surface reconstruction [41] is used to reconstruct the initial meshes. Finally, we apply the method [12] for the texture mapping. To implement our line proposal and building scaffolding algorithm, we take the advantage of OpenCV and CGAL to do image processing and mesh manipulation.

The system is tested on four datasets, two stair-shaped buildings (“stair-shaped 1” and “stair-shaped 2”), a monument tower (“monument”), and a capitol-like building (“capitol”). The statistics on the data and the user interaction efforts are listed in Table 2. The scaffold structures of all experimental datasets are built interactively in five minutes on average. With the help of the proposal algorithm, only about 4% lines are manually added, showing that our system greatly reduces the required line drawing efforts. Also note that the relations of the drawn and proposed lines are automatically inferred by our scaffold topological optimization algorithm, so careful line drawing is not necessary. The initial meshes, consolidated scaffolds, and final results are shown in Fig. 11. “stair-shaped 1” and “stair-shaped 2” are two typical buildings suitable for scaffolding. However, it is worth noting that the dome roof of “capitol” and the trees in all examples are well preserved. In contrast, such non-planar structures are over-fitted and distorted in primitive-fitting approaches as shown in Fig. 12.

We further demonstrate how the mesh enhanced by our scaffold structures improves the results in conventional applications of mesh simplification and non-photorealistic rendering (NPR). Fig. 13 indicates the comparison of the results of the mesh simplification method [2] after applying different mesh enhancement techniques. Fig. 12 coincidentally demonstrates that automatic primitive fitting [3] does not work well on our datasets due to large noise in the initial meshes, while the simplified results after performing the proposed regularization method preserve the building structure well. As shown in Fig. 14, the mesh enhanced by our scaffold structure can be rendered using a line sketching style [42] more faithfully.

We also evaluate our method on defect laden data. As shown in Fig. 15, the first two meshes suffer heavy noise, and the last mesh has incomplete geometry. The noise is added by displacing the vertices with random vectors whose norms are bounded by 1% or 4% of the bounding box. We can see that the result with 1% noise recovers an almost perfect building structure after the regularization, and that the result with 4% noise still preserves the base structure, although some parts of the detailed geometry are less accurate. The geometric incompleteness of the reconstructed mesh, such as holes and concaves, is caused by non-Lambertian materials or object occlusion. Note that at the preprocessing step, holes are filled with flat surfaces, shown in the third row of Fig. 15. We find that when the incompleteness is not significant, the linear features are still nicely recovered (the long lines in the blue region). However, our method may not work well in the presence of relatively large incompleteness (the short lines in the red region).

Limitations. Since the 3D linear features are obtained by back-projecting 2D lines onto the mesh, the camera geometry and input mesh affect the precision of extracted scaffold lines. Given a high level of noise or incompleteness, the

TABLE 2
Statistics on the experiments.

Dataset	Mesh	View selection		Lines					Interaction	
	#Faces	#Select	#Total	#Automatic	# User drawn	#Proposed	# Deleted	# Total	Ratio	Time (min)
stair-shaped 1	864557	5	192	479	28	298	6	799	0.035	3
stair-shaped 2	872571	4	153	177	8	52	4	233	0.034	2.1
capitol	1980439	6	93	481	37	368	21	865	0.043	5.5
monument	1075487	5	157	276	14	90	4	379	0.037	2.6

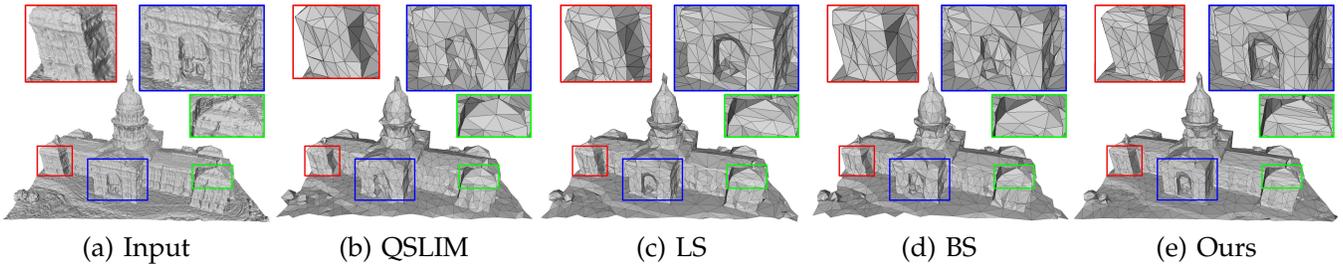


Fig. 13. Comparison of the improvements for mesh simplification. (a) Input mesh. (b) QSLIM: result of QSLIM. (c) LS: result of QSLIM after Laplacian smoothing. (d) BS: result of QSLIM after bilateral denoising. (e) Ours: result of QSLIM after our regularization.

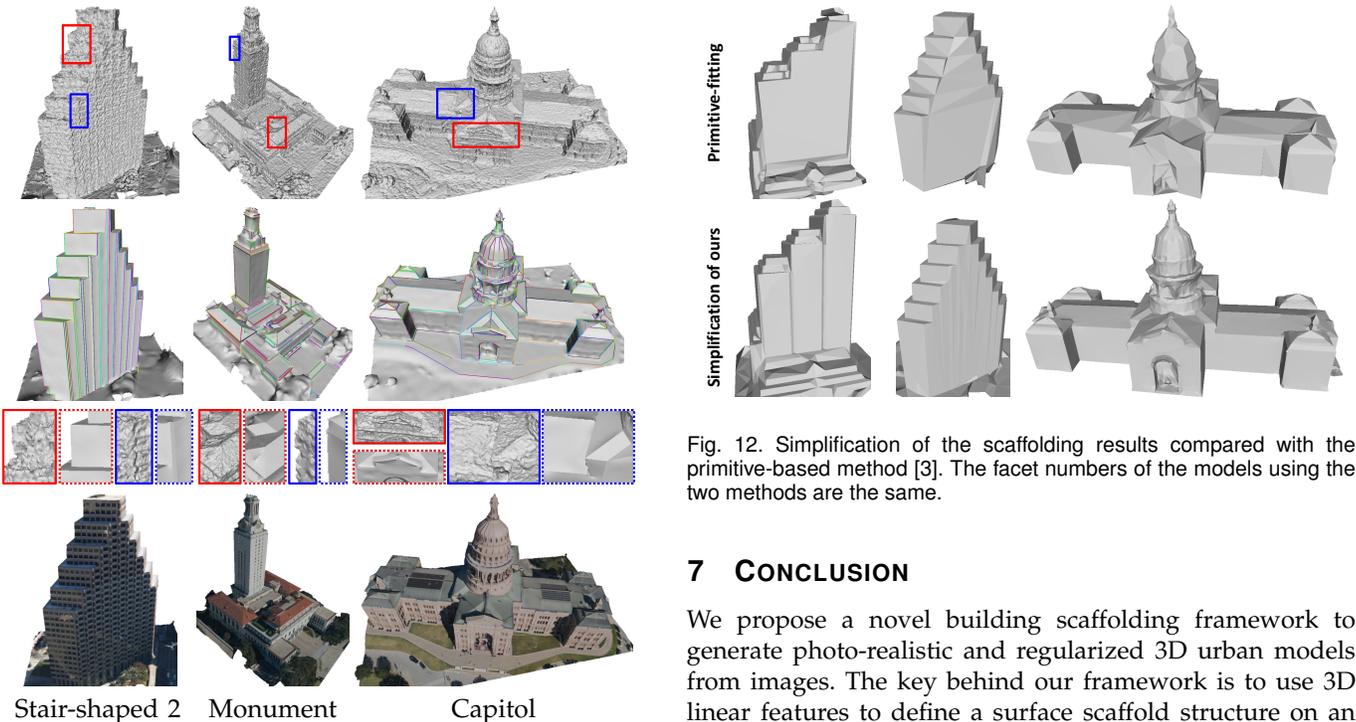


Fig. 12. Simplification of the scaffolding results compared with the primitive-based method [3]. The facet numbers of the models using the two methods are the same.

7 CONCLUSION

We propose a novel building scaffolding framework to generate photo-realistic and regularized 3D urban models from images. The key behind our framework is to use 3D linear features to define a surface scaffold structure on an existing mesh instead of dividing the original mesh into a few artificial primitives. Such a structure only constrains sharp and salient features on the mesh, while the remaining regions are assumed to be smooth. This property makes our scaffold structure very flexible for regularizing not only planar structures, but also other arbitrary surfaces. The embracing of lines allows users to sketch lines, which is the most natural way of interaction, to gradually enhance the 3D models. The line proposal algorithm learns and proposes the lines needed in the regularization according to the user-drawn lines, which greatly reduces tedious manual intervention. The scaffold consolidation further automatically and robustly infers complete spatial relations between the lines that may be sparse, incomplete and conflicting and

Fig. 11. Final results. From top to bottom: initial meshes, consolidated scaffold overlaid on the regularized meshes, close-ups of the regularized meshes and textured regularized meshes.

initially proposed scaffold lines may suffer large errors in poses. Our topology optimization approach (Algorithm 2) that starts from a severely bad initial guess may produce an unsatisfactory result (the red regions in the second and third row of Fig. 15). Thus, if the level of noise or incompleteness overwhelms the level of the features, those features may not be precisely recovered, nor the LPW relations in the surrounding regions.

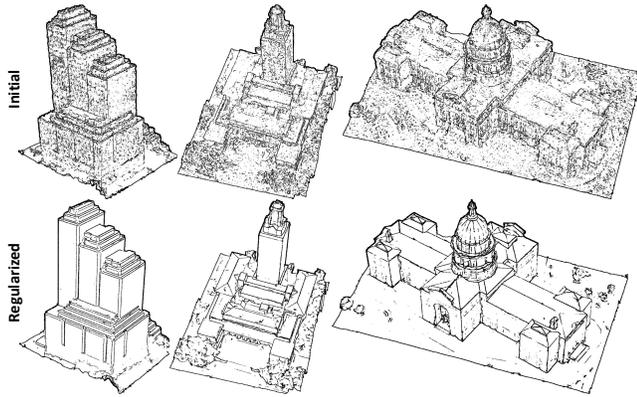


Fig. 14. NPR application.

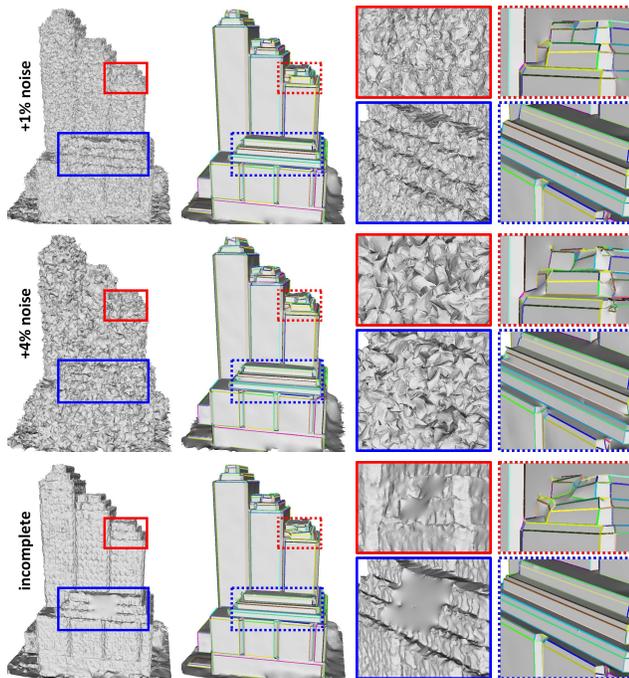


Fig. 15. Regularization on noisy and incomplete data. From left to right are inputs, regularized results, close-ups of inputs and results. The result of 1% noisy data remains high quality in terms of the recovered geometry and LPW relation labeling, while the result of 4% noise is less accurate in the detailed structure (red region). There are two regions missing structure in the incompleteness data. Lines longer (in blue region) than the incomplete region can be nicely recovered while shorter lines (in red region) cannot.

optimize the positional structure of the scaffold to achieve model regularization. Experiments have shown that our systems can help users to regularize the building meshes generated by any automatic image-based reconstruction approaches and significantly improve the results of subsequent processing such as simplification and NPR rendering.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive comments, Zhaohua Li for her help in the early version of the paper, Zhexi Wang and Shiwei Li for their valuable discussion, and Shauna Dalton for proofreading the final version. This work was supported

by RGC-GRF 16208614, 618711, 16209514, 619611, and ITC-PSKL12EG02.

REFERENCES

- [1] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 27, no. 3, pp. 418–433, 2005.
- [2] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Annual Conference on Computer Graphics and Interactive Techniques*, 1997, pp. 209–216.
- [3] A.-L. Chauve, P. Labatut, and J.-P. Pons, "Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [4] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 8, pp. 1362–1376, 2010.
- [5] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [6] H.-H. Vu, P. Labatut, J.-P. Pons, and R. Keriven, "High accuracy and visibility-consistent dense multiview stereo," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, no. 5, pp. 889–901, 2012.
- [7] Y. Zheng, H. Fu, O.-C. Au, and C.-L. Tai, "Bilateral normal filtering for mesh denoising," *Transactions on Visualization and Computer Graphics (TVCG)*, vol. 17, no. 10, pp. 1521–1530, 2011.
- [8] L. He and S. Schaefer, "Mesh denoising via l0 minimization," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 64, 2013.
- [9] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," in *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3. ACM, 2003, pp. 950–953.
- [10] F. Lafarge and P. Alliez, "Surface reconstruction through point set structuring," *Computer Graphics Forum*, vol. 32, no. 2, pp. 225–234, 2013.
- [11] D. Gallup, J.-M. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [12] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys, "Interactive 3d architectural modeling from unordered photo collections," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5. ACM, 2008, p. 159.
- [13] J. Xiao, T. Fang, P. Tan, P. Zhao, E. Ofek, and L. Quan, "Image-based façade modeling," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5, pp. 161:1–161:10, Dec. 2008.
- [14] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach," in *Conference on Computer Graphics and Interactive Techniques*, 1996.
- [15] A. van den Hengel, A. Dick, T. Thormählen, B. Ward, and P. H. Torr, "Videotrace: rapid interactive scene modelling from video," in *ACM Transactions on Graphics (TOG)*. ACM, 2007.
- [16] A. Jain, C. Kurz, T. Thormählen, and H.-P. Seidel, "Exploiting global connectivity constraints for reconstruction of 3d line segments from images," in *Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [17] S. Ramalingam and M. Brand, "Lifting 3d manhattan lines from a single image," pp. 497–504, 2013.
- [18] S. N. Sinha, D. Steedly, and R. Szeliski, "Piecewise planar stereo for image-based rendering," in *International Conference on Computer Vision (ICCV)*, 2009, pp. 1881–1888.
- [19] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Manhattan-world stereo," in *Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [20] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik *et al.*, "Building rome on a cloudless day," in *European Conference on Computer Vision (ECCV)*, 2010.
- [21] Y. Li, X. Wu, Y. Chrysathou, A. Sharf, D. Cohen-Or, and N. J. Mitra, "Globfit: Consistently fitting primitives by discovering global relations," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 4. ACM, 2011, p. 52.
- [22] J. Chen and B. Chen, "Architectural modeling from sparsely scanned range data," *International Journal on Computer Vision (IJCV)*, vol. 78, no. 2-3, pp. 223–236, 2008.

- [23] F. Lafarge and C. Mallet, "Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation," *International Journal of Computer Vision (IJCV)*, vol. 99, no. 1, pp. 69–85, 2012.
- [24] R. Schnabel, P. Degener, and R. Klein, "Completion and reconstruction with primitive shapes," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 503–512.
- [25] M. Arikian, M. Schwärzler, S. Flöry, M. Wimmer, and S. Maierhofer, "O-snap: Optimization-based snapping for modeling architecture," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 1, pp. 6:1–6:15, 2013.
- [26] Y. Zhuang, M. Zou, N. Carr, and T. Ju, "A general and efficient method for finding cycles in 3d curve networks," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 180, 2013.
- [27] L. Yang, J. Liu, X. Tang *et al.*, "Complex 3d general object reconstruction from line drawings," 2013.
- [28] T. K. Dey, K. Li, and J. Sun, "On computing handle and tunnel loops," in *Cyberworlds, 2007. CW'07. International Conference on. IEEE*, 2007, pp. 357–366.
- [29] S. Zhu, T. Fang, R. Zhang, and L. Quan, "Multi-view geometry compression," in *Asian Conference on Computer Vision (ACCV)*, 2014.
- [30] R. Rojas, "Adaboost and the super bowl of classifiers a tutorial introduction to adaptive boosting," *Freie University, Berlin, Tech. Rep*, 2009.
- [31] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: a line segment detector," *Image Processing On Line*, 2012.
- [32] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence (PAMI)*, no. 6, pp. 679–698, 1986.
- [33] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence (PAMI)*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [34] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [35] Y. Kurata and M. J. Egenhofer, "The head-body-tail intersection for spatial relations between directed line segments," in *Geographic Information Science*. Springer, 2006, pp. 269–286.
- [36] M. P. Kumar, P. Ton, and A. Zisserman, "Obj cut," in *Computer Vision and Pattern Recognition (CVPR)*, 2005.
- [37] O. Sorkine and D. Cohen-Or, "Least-squares meshes," in *Proceedings of the Shape Modeling International 2004*, ser. SMI '04. IEEE Computer Society, 2004, pp. 191–199.
- [38] G. Noris, A. Hornung, R. W. Sumner, M. Simmons, and M. Gross, "Topology-driven vectorization of clean line drawings," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 1, p. 4, 2013.
- [39] Z. Wang, F. Wu, and Z. Hu, "Msls: A robust descriptor for line matching," *Pattern Recognition*, vol. 42, no. 5, pp. 941–953, 2009.
- [40] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [41] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Eurographics symposium on Geometry processing*, 2006.
- [42] J. Liu, H. Fu, and C.-L. Tai, "Dynamic sketching: Simulating the process of observational drawing," *CAE'14 Proceedings*, 2014.



dering.

Tian Fang received the bachelor and master degree in Computer Science and Engineering from the South China University of Technology, China, in 2003 and 2006, respectively, and received the Ph.D. degree in Computer Science and Engineering from the Hong Kong University of Science and Technology (HKUST) in 2011. He is now a research assistant professor in HKUST. His research interests include large scale image-based modeling, mesh vectorization, image segmentation, recognition, and photo-realistic ren-



Qingkun Su is a PhD student in the Hong Kong University of Science and technology. He received his BS degree on software engineering from Shandong University in 2012. His research interests include computer graphics, human-computer interaction and image processing.



Siyu Zhu received the bachelor degree in Computer Science from Zhejiang University, China, in 2012, and he is a third year Ph.D. candidate in the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology (HKUST). His research interests include computer vision and computer graphics, especially large-scale 3D reconstruction.



Jingbo Liu is currently a Ph.D candidate in Computer Science at the Hong Kong University of Science and Technology (HKUST). He received his B.Eng. from HKUST, double-majoring in Computer Science and Electronic Engineering. His research interests include computer graphics and human-computer interaction.



Jinglu Wang is a PhD student in the Hong Kong University of Science and technology. She received her BS degree on Computer Science and Technology from Fudan University in 2011. Her research interests include 3D reconstruction, image-based modeling and scene parsing.



Shengnan Cai received the bachelor's degree in software engineering from Shandong University, China, in 2012. She is now a PhD student from the Hong Kong University of Science and Technology (HKUST). Her research interests include image parsing, object recognition and tracking, fashion search and matching.



Chiew-Lan Tai received the BSc degree in mathematics from the University of Malaya, the MSc degree in computer and information sciences from the National University of Singapore, and the DSc degree in information science from the University of Tokyo. She is currently a professor of computer science at the Hong Kong University of Science and Technology. Her research interests include geometry processing, computer graphics, and interactive techniques.



Long Quan is a Professor of the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology (HKUST). He received his Ph.D. in 1989 in Computer Science from INPL, France. He entered as a permanent researcher into the Centre National de la Recherche Scientifique (CNRS) in 1990 and was appointed at the Institut National de Recherche en Informatique et Automatique (INRIA) in Grenoble, France. He joined the HKUST in 2001, and was the founding Director

of the HKUST Center for Visual Computing and Image Science. He is a Fellow of the IEEE Computer Society. He works on vision geometry, 3D reconstruction and image-based modeling. He supervised the first French Best Ph.D. Dissertation in Computer Science of the Year 1998 (le prix de thèse SPECIF 1998, now le prix de thèse Gilles Kahn), the Piero Zamperoni Best Student Paper Award of the ICPR 2000, and the Best Student Poster Paper of IEEE CVPR 2008. He co-authored one of the six highlight papers of the SIGGRAPH 2007. He was also elected as the HKUST Best Ten Lecturers in 2004 and 2009. He has served as an Associate Editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) and a Regional Editor of Image and Vision Computing Journal (IVC). He is on the editorial board of the International Journal of Computer Vision (IJCV), the Electronic Letters on Computer Vision and Image Analysis (ELCVIA), the Machine Vision and Applications (MVA), and the Foundations and Trends in Computer Graphics and Vision. He was a Program Chair of IAPR International Conference on Pattern Recognition (ICPR) 2006 Computer Vision and Image Analysis, is a Program Chair of ICPR 2012 Computer and Robot Vision, and is a General Chair of the IEEE International Conference on Computer Vision (ICCV) 2011.