Image Parallax based Modeling of Depth-layer Architecture

Yong Hu, Bei Chu, Yue Qi

State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
School of New Media Art and Design, Beihang University

Abstract. We present a method to generate a textured 3D model of architecture with a structure of multiple floors and depth layers from image collections. Images are usually used to reconstruct 3D point cloud or analyze facade structure. However, it is still a challenging problem to deal with architecture with depth-layer structure. For example, planar walls and curved roofs appear alternately, front and back layers occlude each other with different depth values, similar materials, and irregular boundaries. A statistic-based top-bottom segmentation algorithm is proposed to divide the 3D point cloud generated by structure-from-motion (SFM) method into different floors. And for each floor with depth layers, a repetition based depth-layer decomposition algorithm based on parallax-shift is proposed to separate the front and back layers, especially for the irregular boundaries. Finally, architecture components are modeled to construct a textured 3D model utilizing the extracting parameters from the segmentation results. Our system has the distinct advantage of producing realistic 3D architecture models with accurate depth information between front and back layers, which is demonstrated by multiple examples in the paper.

1 Introduction

Realistic and flexible 3D architecture models are very important for many applications including culture heritage protection, games, movies and augmented reality navigation etc. Depth images from 3D scanners or color images from digital cameras are the two most popular data sources to model the architecture. Obviously, digital cameras are more common and inexpensive, and also provide rich color texture information which is very important for realistic modeling. Therefore, we focus our work on the problem of image based architecture modeling.

Many works have been proposed for this problem, e.g. [1] focuses on piecewise planar architecture; [2] utilizes a single image to model symmetry architecture, but it needs a lot of manual interactions; [3] makes use of a rectangular plane or a developable surface to generate buildings. There are also a few commercial tools which all require tedious manual works. However, both of them can not deal with architecture with front and back depth layers and get the accurate

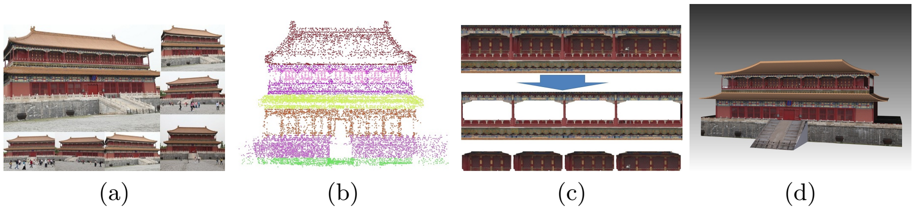


Fig. 1. Our modeling results. (a) Image collections. (b) Top-bottom segmentation result. (c) Depth-layer decomposition result. (d) The final textured 3D model.

depth between these layers automatically. Front-back depth layers are very common in architecture, e.g. in Figure 1 (a), the second floor of this building are composed of two layers which are pillars layer and windows layer respectively. This property is particularly worthy of being modeled, [4] proposes a 2D-3D fusion method to decompose these layers to rectangular planar fragments, and a 3D LiDAR scanner is also needed. We handle this problem only with digital images especially for decomposing layers with irregular boundaries (as shown in the corner regions in Figure 1 (c)) as one of our contributions. Another kind of methods are based on multi-view stereos, such as [5], [6], etc.. The reconstruction results are dense point clouds or meshes and can not be further segmented to generate meaningful architecture components which are more important to extend the range of architecture modeling application.
Given image collections of one building, our goal is to generate a visually compelling 3D model, in which accurate architecture components segmentation is critical. Although high resolution texture information can be acquired from images, but it is hard to segment the components in the image space solely. A common case is that different components which are occluded by each other may have the same material, such as the pillars and their back windows in Figure 1 (a). 3D point cloud can be generated from multiple images by structure-from-motion (SFM) method, but these 3D points are too sparse to be segmented into architecture components directly. The complementary characteristics of these two data sources are combined to handle more complex architecture modeling problems in this paper. We first propose a statistic-based top-bottom segmentation algorithm and divide the sparse 3D point cloud to several horizontal floors vertically along the ground plane normal as in Figure 1 (b). For each horizontal floor with depth layers, we propose a repetition based depth-layer decomposition algorithm to divide its sparse 3D points and images into several repetitive components as in Figure 1 (c). The key observation of our depth-layer decomposition algorithm is the parallax-shift among repetitive structures in a single image or multiple view images. Finally, textured architecture component models are reconstructed from the segmented sparse 3D points and texture parts to make up the 3D architecture model. An example of modeling results is shown in Figure 1 (d), and others are shown in the section Experiments and Discussion, which prove that our work can be used in practical applications.

2 Related works

Image based architecture modeling have received a lot of research interest, with a large spectrum of modeling systems developed to build realistic 3D models. We classify the up to date and most relevant studies according to the data sources, single view or multiple view images without being exhaustive.

Single view based modeling [7] represents a scene as a layered collection of depth images, but assigns depth values and extracts layers manually. [8] presents a fully automatic method for creating a rough 3D model from a single photograph, the model is composed of several texture-mapped planar billboards. [9] automatically extracts shape grammar rules from facade images for procedural modeling technology whose modeling results are similar as ours, but the depth are also assigned manually. [10] makes use of Manhattan structure for man made building and models the building as a number of vertical walls and a ground plane. [2] calibrates the camera and reconstructs a roughly sparse point clouds from a single image by exploiting symmetry, but user must interactively marks out components on the 2D image to complete the modeling work. Realistic textured 3D models are reconstructed but depth layer decomposition are not handled. [11] proposes a repetition-based dense single-view reconstruction method, but the repetition is necessary, and depths are roughly estimated from the repetition intervals.

Multiple views based modeling Image collections from different viewpoints are able to provide more 3D geometric information. Computer vision based multi-view stereo(MVS) algorithms, such as [5] and [6], generate architecture meshes on dense stereo reconstruction method. [12] and [13] develop a real-time video images registration method and focus on the global reconstruction of dense stereo results. Proper modeling of the structure from reconstructed point clouds or meshes has not yet been addressed. Recently, some methods use 3D points from SFM or MVS to guide users for marking out the architecture components interactively and efficiently. [1] uses image collections to assist interactively reconstructing the architecture composed of planar elements. [3] proposes a semi-automatic method to segment the architecture and optimizes a depth value for each component using reconstructed 3D points. The following paper [14] proposes a partition scheme to separate the scene into independent blocks and extends their methods to reconstruct street-side city block. [15] introduces a schematic representation for architecture, in which transport curves and profile curves are extracted from 3D point cloud to generate an architecture model with some swept surfaces. [16] uses baseline and profile, a similar representation of [15], to model the architecture facade in the unwrapped space and get a textured architecture model. But the above methods are not suited for separating the depth layers with irregular boundaries and similar appearance. [4] uses LiDAR data which are manually registered with photos to generate building with multiple layers. However, their layer decomposition method needs denser point cloud than SFM and only deals with rectangular components.

3 Overview

In the real world, most buildings are multistory and also have different depth layers. For the purpose of reconstructing the structure and components of the architecture, we first segment the architecture into isolated floors, and further divide some floors to repetitive or non-repetitive parts, and finally decompose these parts to layers with different depth values. In this paper, floor-segmentation is implemented in a 3D point cloud space generated from multiple view images by SFM algorithm, and we call it top-bottom floors segmentation. Integration of the above 3D points segmentation results and the multiple view images, repetition detection and further segmentation is performed to solve the decomposition problem for the floors with depth-layer structure, and we call it depth-layer decomposition. The pipeline of our architecture modeling method is composed of four major stages.

1) 3D point cloud from SFM From the captured images, a sparse 3D point cloud is reconstructed by SFM method, then outliers removing and normals estimation are performed. The reconstruction results are used as our input for the following segmentation.

2) Top-bottom floors segmentation Manhattan directions[17] are first estimated from the normals of the 3D points. Along the direction which is parallel to the normal of the ground plane, the 3D points are partitioned vertically to different horizontal floors. For some horizontal floors, 3D points are further partitioned to different layers according to their depth values.

3) Depth-layer decomposition For the candidate decomposition floor, 3D points at different layers are projected back to images and help us to detect the horizontal repetition at different layers respectively in the image space. Then, for each repetitive region, per-pixel parallax-shift values are estimated using SIFT-flow method[18], and the region is further decomposed into front and back layers by solving a per-pixel label-assignment optimization problem.

4) Architecture components modeling Parameters are extracted from the corresponding 3D points clusters and their projection images to generate the geometry of the architecture components. Then the components' textures are repaired from the multiple view segmented images, and a textured 3D architecture model is reconstructed finally.

4 Statistic-based Top-Bottom Floors Segmentation

Our top-bottom floors segmentation algorithm is based on two intuitive criteria: normal variation that separates components like roofs and walls, and the depth variation that distinguishes layers with different depth values. Firstly, point cloud is generated by SFM algorithm and preprocessed by outliers removing and normals estimation. Secondly, Manhattan-direction is estimated as our segmentation direction. Finally, 3D points are segmented to floors along the ground normal direction according to the points' normal variation and some floors are further segmented to layers with different depth values along the facade direction.

4.1 Image Capture and Point Cloud Preprocessing

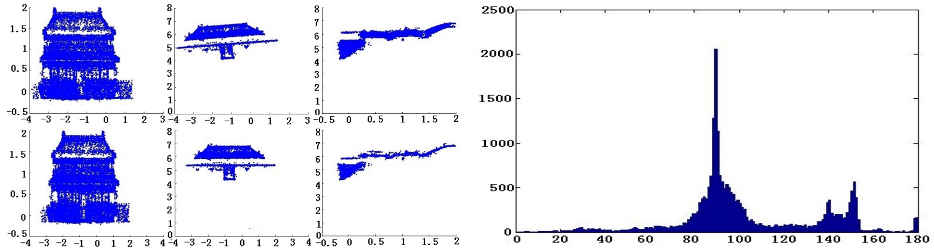
About 100 images for each dataset are captured with the positions distributing on an 180-degree arc in front of the architecture. A 3D point cloud is reconstructed using VisualSFM [19] because of its stability and ease of use. Generally, SFM point cloud contains outliers, which can be removed by performing a radius outlier removing method. In addition, point normals can only be inferred from the point cloud dataset directly. The problem of point normal estimation is approximated by the problem of estimating the normal of a plane tangent to the surface, which in turn becomes a least-square plane fitting estimation problem. For each point, an analysis of the eigenvectors and eigenvalues of a covariance matrix created from the nearest neighbors is implemented to estimate its normal. PCL-Point Cloud Library [20] is used to complete our outliers removing and point normal estimation.

4.2 Manhattan Axes Estimation

The 3D point cloud from SFM is in the camera coordinate system of the primary image as shown in the top three images of Figure 2 (a), which can not be directly used to vertically segment the 3D points into horizontal floors. Therefore, the ground plane normal should be estimated firstly. Many existing methods adapt the Manhattan assumption, estimate the Manhattan-axis in the case of piecewise planar architecture, and generate axis-aligned plane segments. We relax this assumption to non-piecewise planar architecture that includes oblique or curved surfaces like roofs. Let X_M, Y_M, Z_M be the Manhattan axes, and in this paper Y_M and Z_M also represent the ground plane normal direction and the facade normal direction of the architecture respectively. Let X_C, Y_C, Z_C be the coordinate axes of the original 3D point cloud. Since the captured images are taken without much yawing and rolling, the facade normal direction Z_M is nearly perpendicular to Y_C. Therefore, a histogram of angles between all the point normals and Y_C is created as shown in Figure 2 (b). The longest column with horizontal coordinate value near to 90 is selected and the corresponding group of point normals are considered as the candidates to estimate Z_M. Another histogram of angles between candidate points' normals and their mean normal is created repeatedly to remove outlier points until all angles between candidate point normals and the mean normal are less than 2 degree. Z_M is assigned as the mean normal of the remaining candidate points after outliers removing. Then, $X_M = Z_M \times Y_C$ and $Y_M = Z_M \times X_M$. Finally, the 3D point cloud is transformed to the Manhattan axes system as shown in the bottom three images of Figure 2(a).

4.3 Segmentation

Normal variation based segmentation With respect to the artificial architecture, it is intuitive that point normals in the same floor vary slightly or



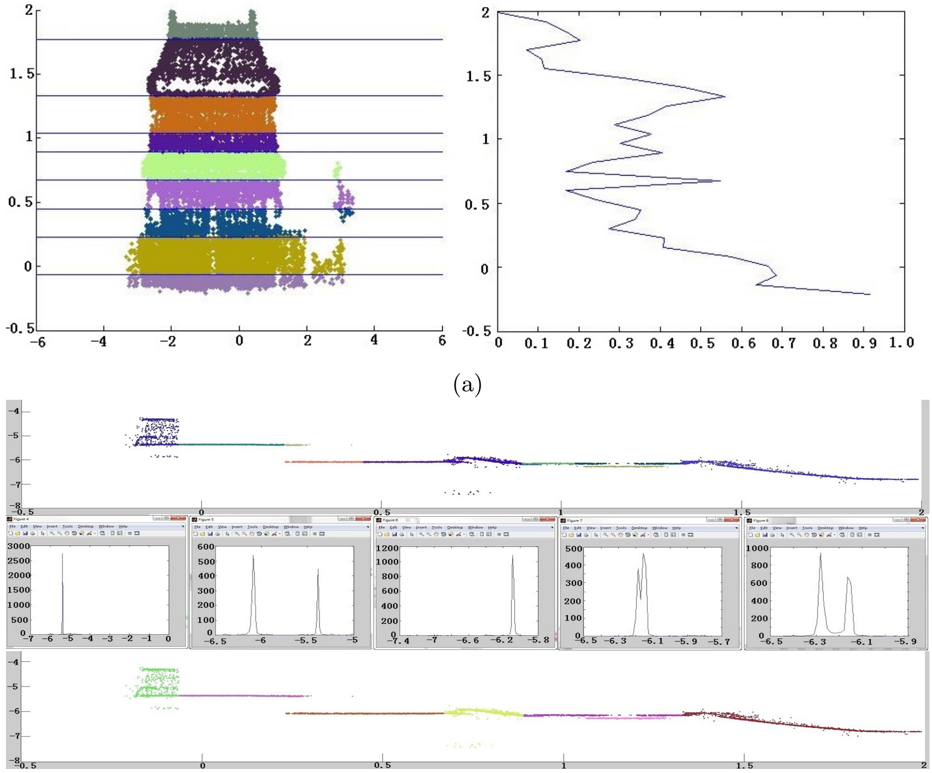
(a) (b)

Fig. 2. (a) Top three images are the three orthographic views of the reconstructed 3D point cloud before Manhattan rectification and bottom three images are the rectification results. (b) The histogram of all the 3D point normals distribution. The vertical axis is the 3D point number and the horizontal axis is the angle between the point normal and ground plane normal.

smoothly. Therefore, we split the 3D point cloud into small slices along Y_M and compute the variance of point normals in each slice. The positions of the local maximum values in the curve composed of the variance values can be considered as the potential split lines to segment the point cloud into different horizontal floors. More specifically, a series of uniform sampling planes which are perpendicular to Y_M are created to divide the 3D point cloud into point slices. In each slice, the variance of the dot product between the point normals and Y_M are computed to form the curve. The interval of these sampling planes are only determined by the ratio of the 3D point cloud height to the real architecture height. Because of the noise points, false split lines will be chosen and result in over-segmentation results. Nonetheless, this over-segmentation will be resolved by the subsequent merging operation. The segmentation result of this step are shown in Figure 3 (a).

Depth-based segmentation In this step, some floors are further divided into separated layers according to the depth values along facade normal direction Z_M. First, roof floors are recognized and removed by the dot products between their mean normals and Y_M. Then, for each remaining floor, a series of uniform sampling planes perpendicular to the axis Z_M are created to divide the current floor into point slices. A statistic curve composed of the numbers of all point slices is built, and the positions of its local maximum values are considered as the split lines along Z_M direction as shown in the middle images of Figure 3 (b). In order to remove the noise, some local maximum values are filtered if the corresponding slices contain less than two percent of all points. The final depth-based segmentation result is shown in the top image of Figure 3 (b).

Merging Due to the presence of noise points, the 3D point cloud will be over segmented by the above two segmentation steps. To solve this problem, any two segmented parts are traversed to determine whether they are able to merge. Two segmented parts with similar normals will be merged if the two distances between their bounding boxes along Y_M and Z_M directions are both less



(a)

(b)

Fig. 3. (a) The result of normal variation based segmentation. Local maximum values of the curve in right image generate the split lines in the left image. (b) The result of depth variation based segmentation. Different segmented parts are represented by different colors in the top image. Statistic curves formed by point numbers of slices for the potential floors to be partitioned are shown in the middle images. Bottom image shows the merging result.

than the sampling plane interval. The above merging operation is implemented repeatedly until there are no mergeable parts and the result is shown in the bottom image of Figure 3 (b).

5 Repetition based Depth-Layer Decomposition

In order to reconstruct the layered and textured 3D architecture model, we should decompose the images into different layers according to their corresponding depth values. During the SFM process, some pairs of 2D pixels and 3D points are established, but they are too sparse to do the direct layer assignment for all pixels. Nevertheless, these sparse pairs can provide enough information for

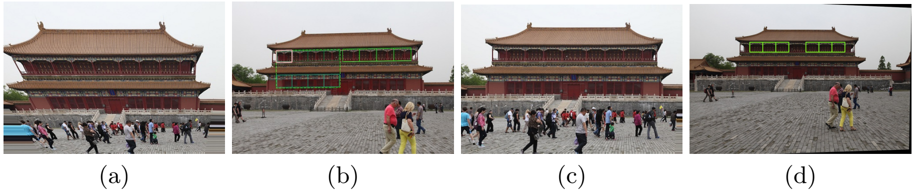


Fig. 4. (a) Image rectification by [22]. (b) Repetition detection by [22]. (c) Image rectification by our method. (d) Repetition detection by our method.

accurate repetition detection in the image with depth-layer structure. Because of the perspective projection and the camera position, image deviation exists among back layer regions behind the front layer repetitive structures, and we name this deviation as *parallax-shift*. A *parallax-shift* estimation based coarse image segmentation algorithm is proposed to perform the initial depth-layer decomposition. Generally, the boundaries between front and back layers are often irregular, but also have a high edge response. Based on above-mentioned characteristic and coarse decomposition result, we design a per-pixel label-assignment formulation and deploy a graph-cut optimization to refine the depth-layer decomposition.

Our repetition based depth-layer decomposition algorithm decomposes the image into components with different depth values in the following stages: (1) Carry out repetition detection in each image to get rectangle repetitive regions. (2) Perform a coarse depth-layer decomposition based on *parallax-shift* estimation between these repetitive regions. (3) Refine the coarse depth-layer decomposition by per-pixel layer assignment using graph-cut[21] energy minimization. (4) Decompose non-repetitive regions with multiple images.

5.1 Repetition Detection

After vertical segmentation, architecture images are segmented to image strips according to different floors. The repetitive structures appear only along the horizontal direction. We first test the method in [22] to detect rectangle repetitive regions, but there exist two problems in our case. First, large roof area and high frequency repetition of tiles always lead to wrong vanish point detection and result in incorrect image rectification (Figure 4 (a)). Second, repetitive structures at different depth layers always affect the estimation of the symmetry axis location (Figure 4 (b)).

In order to satisfy our subsequent decomposition requirement, we improve Wu's method[22] in three ways to achieve higher accuracy and stability. (1) Some 3D points are selected randomly, and the lines across these points and parallel to the Manhattan coordinate axes are computed and projected on the captured images to estimate the vanish points. This improvement utilizes the 3D point cloud information and results in consistent rectification for multiple view images. Meanwhile, wrong rectification results caused by parallel lines which are not vertical or horizontal such as parallel lines on the roof are avoided. (2) According to the former segmentation in point cloud, we are able to pick out the SIFT points at the front layer to estimate the repetitive region size and the symmetry axis location. This improvement may remove the interference by the repetition at different layers. (3) Vector quantization is used to estimate the similarity of different repetitive regions. A quad tree is constructed for the image, the root node is the whole image, and the image region belonging to each child node is one quarter of its parent node. The image region size in the leaf node is half size of the smallest repetitive region. SIFT descriptors are computed for all nodes and clustered to 256 categories (In our experiment, 256 is totally enough to detect the repetitive region similarity). Any two candidate regions are confirmed as repetitive regions if their SIFT descriptors belong to the same category. This improvement avoids the sensitive threshold value determination in Wu's method[22]. Our image rectification and repetition detection results are shown in Figure 4 (c) and (d).

5.2 Coarse Depth-layer Segmentation

For the rectified image, the camera projection plane is parallel to the architecture facade. The parallax-shift between the projection pixels of two points with the same horizontal coordinates but different depth values can be computed according to

$$ps(x) = x \times (d_2 - d_1)/d_1 d_2 = Cx \tag{1}$$

as in Figure 5.

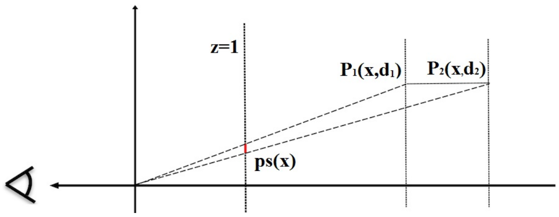


Fig. 5. The principle of parallax-shift between two points which is represented by the red solid line segment $ps(x)$.

Convert to the repetitive regions in the image, repeating points at the back layer project to different pixel locations in respective repeating regions (Figure 6 (a)). More specifically, given a set of repeating regions $\{R_1, R_2, .., R_n\}$ with symmetry axes $\{X_1, X_2, .., X_n\}$, for any point P_i at the front layer and its corresponding point P_i' at the back layer in R_i, its corresponding repetitive point in R_j is denoted as P_j and P_j', they satisfy the following equations:

$$I_x(P_i) - X_i = I_x(P_j + (t_{ij}, 0)) - X_j, \tag{2}$$

where $I_x(P_i)$ is the x coordinate of the projection of point P_i on image I, t_{ij} is the distance between R_i and R_j.

$$I_x(P_i') - X_i = I_x(P_j' + (t_{ij}, 0)) - X_j + ps_{ij}, \tag{3}$$

where $ps_{ij} = t_{ij} \times C$ is the parallax-shift between R_i and R_j.

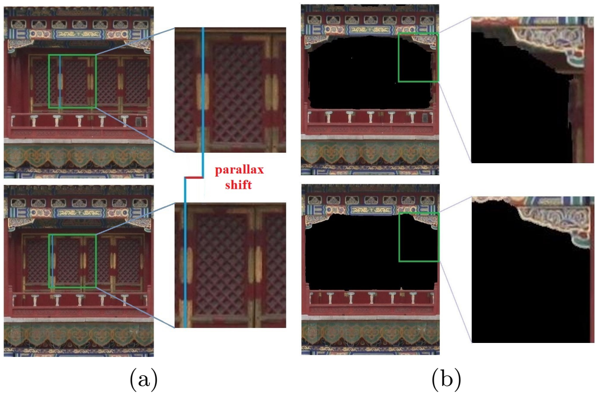
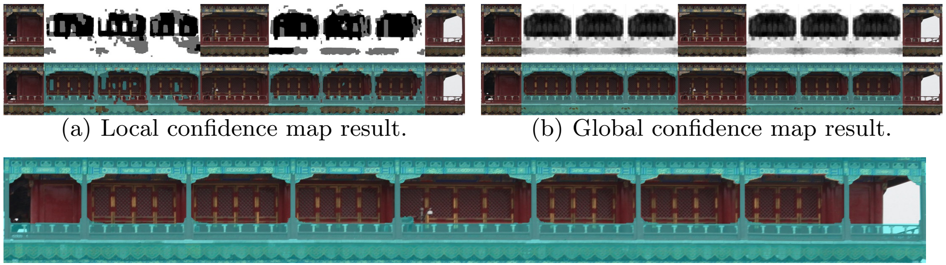


Fig. 6. (a) The blue lines are repeating back layer structures in different repeating regions. The parallax-shift is represented by a short red line. (b) Depth-layer decomposition results for one repetitive region. Part of the image is extracted to show details.

Given two repetitive regions, SIFT-Flow method is used to compute the parallax-shift and obtain a flow vector map. The horizontal component of the flow vector is a coarse indicator of the corresponding pixel's layer label, but it is very unfaithful. Therefore, flow vector maps between multi-pair repetitive regions are computed to estimate a consistent confidence map for the repetitive regions. The confidence map is used as the input for the graph-cut optimization method to refine the depth-layer decomposition in the next section. Our confidence map calculation includes two stages. (1) Local confidence map calculation. For each region R_i, the flow vector maps between it and R_{i-2}, R_{i-1}, R_{i+1}, R_{i+2} are computed respectively. These flow vector maps are converted to confidence maps as the following equation.

$$cm_{ij}(s) = \begin{cases} 1 \quad \text{if} \;\; |(h_{i,j}(s)/t_{ij}| < C/5 \\ -1 \text{ else} \end{cases} \tag{4}$$

where $cm_{ij}(s)$ denotes the confidence map contributed from R_j to R_i, $h_{ij}(t)$ denotes the horizontal component of the flow vector map between R_i and R_j. Local confidence map of R_i is calculated by accumulating all the confidence maps from neighboring repetitive regions (Figure 7 (a)). (2) Global confidence map calculation. For a group of repetitive regions, only one uniform confidence map is calculated by summing all the regions' local confidence maps. During the summation of local confidence maps, the axial and translational symmetry are also considered to increase the number of votes and improve the robustness of the global confidence map (Figure 7 (b)).



(a) Local confidence map result.

(b) Global confidence map result.

(c) Final decomposition result.

Fig. 7. For (a) and (b), top image shows the confidence value directly. Bottom image shows the segmentation results by blending the confidence value and pixel color value. (c) Final decomposition result including non-repetitive regions.

5.3 Decomposition Refinement

After coarse depth-layer decomposition, a uniform decomposition result for each repetitive region is obtained and shown as in the top image of Figure 6 (b). However, the boundary of the decomposition result is not accurate. In order to refine it, local edge information is utilized to optimize the boundary to the pixels with maximal gradient variation. Inspired by the interactive Graph-Cut method [21], a similar Markov Random Field (MRF) energy function is constructed, global confidence map for each repetitive region is assigned as the data term instead of the interactive constrain in [21], and the pixel edge response is assigned as the smooth term as the following equations.

$$E(L) = \sum E_{data}(L(s), s) + \sum_{L(p)!=L(q)} E_{smooth}(p, q) \tag{5}$$

where L denotes the layer label map of a repetitive region. There are only two kinds of values in L: 1 means front layer and 0 means back layer. s denotes one pixel in a repetitive region, (p, q) denotes one pair of neighboring pixels.

$$E_{data}(L(s), s) = \frac{1}{\sigma} e^{-cm'(L(s),s)} \tag{6}$$

$$E_{smooth}(p, q) = \frac{1}{dist(p, q)} e^{-\frac{edge(p)+edge(q)}{\gamma}} \tag{7}$$

edge is the canny edge response of the region image, $edge(p) = 1$ means p is a edge pixel, or else $edge(p) = 0$, and γ is the smooth factor. The data term in the energy function is constructed by the confidence maps from coarse decomposition as:

$$cm'(L(s), s) = \begin{cases} K & \text{if } \; L(s) = 1\&cm(s) > 0 \\ K & \text{if } \; L(s) = 0\&cm(s) < 0 \\ -|cm(s)| & \text{else} \end{cases} \tag{8}$$

where K is set to 9 in the case of 2D image due to the constructed eight-connected graph as in method [21]. Equation 5 is optimized by a graph-cut algorithm [23] and the final refined result is shown in the bottom image of Figure 6 (b).

5.4 Non-Repetitive Region Decomposition

Some repetitive regions over a wide distance are very difficult to detected (Left region and right region in Figure 7 (a) & (b)), and there also exist some non-repetitive regions (Center region in Figure 7 (a) & (b)). Fortunately, our depth-layer decomposition algorithm in a single image can be easily extended to multiple view images with approximate camera parameters. The remaining non-repetitive regions are decomposed by utilizing multiple view images and the results are shown in Figure 7 (c).

6 Experiments and Discussion

6.1 Architecture Geometry Modeling

After segmentation, we get architecture components composed of 3D points and texture images. In order to construct a 3D textured architecture model, textured plane models are used to fit planar components, and parametric surface models to fit non-planar components. The boundary of each component's sparse 3D points is projected onto the images to get a coarse texture boundary. Along the texture boundary, image windows with 100-pixel width are created, where pixels with maximum gradient variation are selected to refine the boundary and extract the modeling parameters. For example, roof is modeled by a quadric surface which is fitted by the parameters (*topwidth*, *bottomwidth*, *height* and *depth*) extracted from segmented texture and 3D points as in Figure 8. The back layer of the architecture is usually occluded by the front layer, textures need to be repaired from multi-view images. For a planar object in 3D space, there exists an affine homography between each view. Image from front-parallel view is chosen as the reference to estimate the affine transforms with other views. The texture holes in this reference image are repaired by warping images from other views with the computed affine transforms. The modeling result of the floor with depth layers is shown in Figure 8.

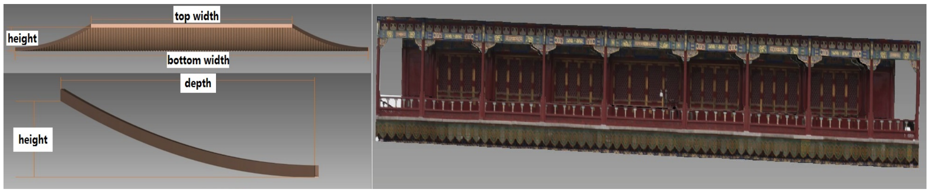


Fig. 8. Left images show the roof modeling parameters and result. Right image shows the depth-layer modeling result.

6.2 Implementation and Results

We demonstrate the results of our approach with three data sets: the HongYi Ge, the Chairman Mao Memorial Hall as shown in Figure 9 and the Hall of Central Harmony as shown in Figure 10.

We first evaluate our method on the ancient Chinese architecture HongYi Ge (Top row in Figure 9). The resolution of our photos is about twenty million pixels. By using continuous shooting mode of the digital camera, these photos are taken in a few minutes for each architecture. VisualSFM[19] is used to generate 3D point cloud as shown in the first column. The result of top-bottom floors segmentation is shown in the second column. The building is segmented successfully into curved roofs and planar facades. For the facade floors with depth layers, repetition based depth-layer decomposition is implemented, and the result is shown in the third column. In this data set, the segmentation on the second floor is very difficult due to the same appearance between the front pillars and the back layer. However, by using both the parallax-shift and edge response, we can get excellent segmentation results. The textured 3D model of this building is shown in the fourth column of Figure 9. Our method also works well on regular modern architectures. In these cases, top-bottom segmentation does not always output meaningful parts, and the output layers are all treated as simple planes as shown in the second row in Figure 9. We also evaluate our method on the architecture without repetitive regions at the front layer as in Figure 10. The modeling result proves that our depth-layer decomposition algorithm can also be applied to non-repetitive architecture.

Limitations As most feature-based methods, SFM result is poor on non-Lambert material and textureless areas. If there exist gaps along the ground plane normal direction, our top-bottom segmentation algorithm will fail. The sparsity of SIFT features may also affect our parallax-shift detection algorithm, which is important in our depth-layer decomposition algorithm based on SIFT-Flow.

7 Conclusion and Future Work

We have presented an image-based modeling approach for architecture with a structure of multiple floors and depth layers. Repetition detection in the image

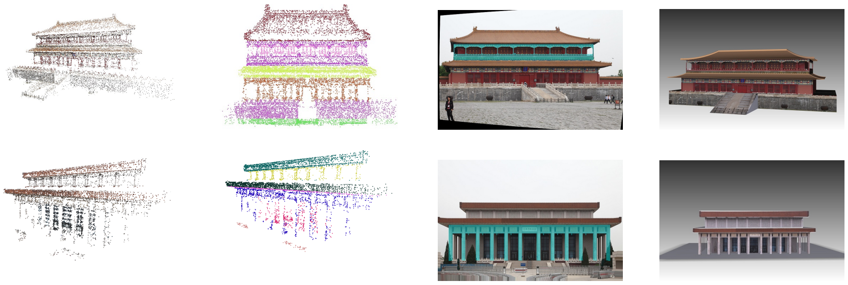
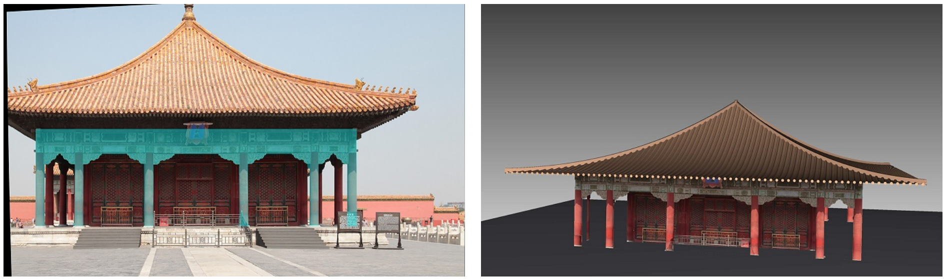


Fig. 9. Left to right: 3D SFM points, top-bottom segmentation, depth-layer decomposition, final reconstructed model.



(a) Depth-Layer decomposition result. (b) Final modeling result.

Fig. 10. Modeling results of Hall of Central Harmony.

region where repetition interference at different layers exists and irregular architecture components decomposition are handled well in our method.
The possible future work includes several directions. The geometry and texture of the architecture components are reconstructed simply which can be further refined to get geometry details and appearance from zoom-in images and make the 3D architecture model relightable. Procedural rules can be extracted to get an editable procedural 3D architecture model and generate large architecture scene quickly.

Acknowledgement.This work was supported in part by ”National Natural Science Foundation of China (NSFC) 61202235, 61272348”, ”Humanity and Social Science Youth foundation of Ministry of Education of China 11YJCZH064”, ”Ph.D. Program Foundation of Ministry of Education of China (no.20111102110018)”, 863 Plan ”2013AA013803”, and ”YWF-13-D2-JC-20”.

References

1. Sinha, S.N., Steedly, D., Szeliski, R., Agrawala, M., Pollefeys, M.: Interactive 3d architectural modeling from unordered photo collections. In: ACM SIGGRAPH

Asia 2008 papers, ACM (2008) 159:1–159:10

2. Jiang, N., Tan, P., Cheong, L.F.: Symmetric architecture modeling with a single image. In: ACM SIGGRAPH Asia 2009 papers, ACM (2009) 113:1–113:8
3. Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E., Quan, L.: Image-based facade modeling. In: ACM SIGGRAPH Asia 2008 papers, ACM (2008) 161:1–161:10
4. Li, Y., Zheng, Q., Sharf, A., Cohen-Or, D., Chen, B., Mitra, N.J.: 2d-3d fusion for layer decomposition of urban facades. In: Proceedings of the 2011 International Conference on Computer Vision, IEEE Computer Society (2011) 882–889
5. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.: Multi-view stereo for community photo collections. In: Proceedings of the 2007 International Conference on Computer Vision. (2007) 1–8
6. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building rome in a day. In: Proceedings of the 2009 International Conference on Computer Vision. (2009) 72–79
7. Oh, B.M., Chen, M., Dorsey, J., Durand, F.: Image-based modeling and photo editing. In: ACM SIGGRAPH 2001 Papers, ACM (2001) 433–442
8. Hoiem, D., Efros, A.A., Hebert, M.: Automatic photo pop-up. In: ACM SIGGRAPH 2005 Papers, ACM (2005) 577–584
9. Müller, P., Zeng, G., Wonka, P., Van Gool, L.: Image-based procedural modeling of facades. In: ACM SIGGRAPH 2007 papers, ACM (2007)
10. Barinova, O., Konushin, V., Yakubenko, A., Lee, K., Lim, H., Konushin, A.: Fast automatic single-view 3-d reconstruction of urban scenes. In: Proceedings of the 10th European Conference on Computer Vision, Springer-Verlag (2008) 100–113
11. Wu, C., Frahm, J., Pollefeys, M.: Repetition-based dense single-view reconstruction. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society (2011) 3113–3120
12. Pollefeys, M., Nistér, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3d reconstruction from video. Int. J. Comput. Vision **78** (2008) 143–167
13. Cornelis, N., Leibe, B., Cornelis, K., Gool, L.: 3d urban scene modeling integrating recognition and reconstruction. Int. J. Comput. Vision **78** (2008) 121–141
14. Xiao, J., Fang, T., Zhao, P., Lhuillier, M., Quan, L.: Image-based street-side city modeling. In: ACM SIGGRAPH Asia 2009 papers, ACM (2009) 114:1–114:12
15. Wu, C., Agarwal, S., Curless, B., Seitz, S.M.: Schematic surface reconstruction. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society (2012) 1498–1505
16. Fang, T., Wang, Z., Zhang, H., Quan, L.: Image-based modeling of unwrappable facades. IEEE Transactions on Visualization and Computer Graphics **19** (2013) 1720–1731
17. Coughlan, J.M., Yuille, A.L.: Manhattan world: Compass direction from a single image by bayesian inference. In: Proceedings of the 1999 International Conference on Computer Vision, IEEE Computer Society (1999) 941–
18. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applicationn. Pattern Analysis and Machine Intelligence **33** (2011) 978–994
19. Wu, C., Agarwal, S., Curless, B., Seitz, S.: Multicore bundle adjustment. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. (2011) 3057–3064
20. Rusu, R., Cousins, S.: 3d is here: Point cloud library (pcl). In: 2011 IEEE International Conference on Robotics and Automation (ICRA). (2011) 1–4

21. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In: Proceedings of the 2001 International Conference on Computer Vision. Volume 1. (2001) 105–112 vol.1
22. Wu, C., Frahm, J.M., Pollefeys, M.: Detecting large repetitive structures with salient boundaries. In: Proceedings of the 11th European conference on Computer vision, Springer-Verlag (2010) 142–155
23. Fulkerson, B., Vedaldi, A., Soatto, S.: Class segmentation and object localization with superpixel neighborhoods. In: Proceedings of the 2009 International Conference on Computer Vision. (2009)