Image-based Façade Modeling

Jianxiong Xiao Tian Fang Ping Tan* Peng Zhao Eyal Ofek† Long Quan

The Hong Kong University of Science and Technology *National University of Singapore †Microsoft

Figure 1: *A few façade modeling examples from the two sides of a street with 614 captured images: some input images in the bottom row, the recovered model rendered in the middle row, and three zoomed sections of the recovered model rendered in the top row.*

Abstract

We propose in this paper a semi-automatic image-based approach to façade modeling that uses images captured along streets and relies on structure from motion to recover camera positions and point clouds automatically as the initial stage for modeling. We start by considering a building façade as a flat rectangular plane or a developable surface with an associated texture image composited from the multiple visible images. A façade is then decomposed and structured into a Directed Acyclic Graph of rectilinear elementary patches. The decomposition is carried out top-down by a recursive subdivision, and followed by a bottom-up merging with the detection of the architectural bilateral symmetry and repetitive patterns. Each subdivided patch of the flat façade is augmented with a depth optimized using the 3D points cloud. Our system also allows for an easy user feedback in the 2D image space for the proposed decomposition and augmentation. Finally, our approach is demonstrated on a large number of façades from a variety of street-side images.

CR Categories: I.3.5 [Computer Graphics]: Computational geometry and object modeling—Modeling packages; I.4.5 [Image Processing and computer vision]: Reconstruction.

Keywords: Image-based modeling, building modeling, façade modeling, city modeling, photography.

1 Introduction

There is a strong demand for the photo-realistic modeling of cities for games, movies and map services such as in Google Earth and Microsoft Virtual Earth. However, most work has been done on large-scale aerial photography-based city modeling. When we zoom to ground level, the viewing experience is often disappointing, with blurry models with few details. On the other hand, many potential applications require street-level representation of cities, where most of our daily activities take place. In term of spatial constraints, the coverage of ground-level images is close-range. More data need to be captured and processed. This makes street-side modeling much more technically challenging.

The current state of the art ranges from pure synthetic methods such as artificial synthesis of buildings based on grammar rules [Müller et al. 2006], 3D scanning of street façades [Früh and Zakhor 2003], to image-based approaches [Debevec et al. 1996]. Müller et al. [2007] required manual assignment of depths to the façade as they have only one image. However, we do have information from the reconstructed 3D points to automatically infer the critical depth of each primitive. Früh and Zakhor [2003] required tedious 3D scanning, while Debevec et al. [1996] proposed the method for a small set of images that cannot be scaled up well for large scale modeling

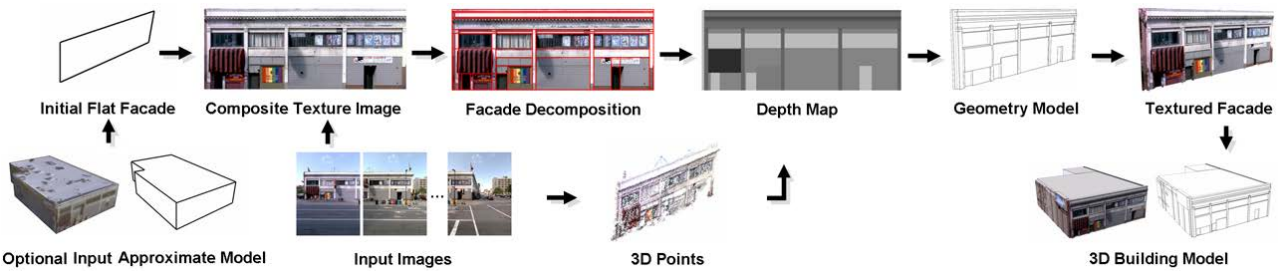


Figure 2: *Overview of the semi-automatic approach to image-based façade modeling.*

of buildings.

We propose a semi-automatic method to reconstruct 3D façade models of high visual quality from multiple ground-level street-view images. The key innovation of our approach is the introduction of a systematic and automatic decomposition scheme of façades for both analysis and reconstruction. The decomposition is achieved through a recursive subdivision that preserves the architectural structure to obtain a Directed Acyclic Graph representation of the façde by both top-down subdivision and bottom-up merging with local bilateral symmetries to handle repetitive patterns. This representation naturally encodes the architectural shape prior of a façade and enables the depth of the façade to be optimally computed on the surface and at the level of the subdivided regions. We also introduce a simple and intuitive user interface that assists the user to provide feedback on façade partition.

2 Related work

There is a large amount of literature on façade, building and architectural modeling. We classify these studies as rule-based, image-based and vision-based modeling approaches.

Rule-based methods. The procedural modeling of buildings specifies a set of rules along the lines of L-system. The methods in [Wonka et al. 2003; Müller et al. 2006] are typical examples of procedural modeling. In general, procedural modeling needs expert specifications of the rules and may be limited in the realism of resulting models and their variations. Furthermore, it is very difficult to define the needed rules to generate exact existing buildings.

Image-based methods. Image-based methods use images as guide to generate models of architectures interactively. Façade developed by Debevec et al. [1996] is a seminal work in this category. However, the required manual selection of features and the correspondence in different views is tedious, and cannot be scaled up well. Müller et al. [2007] used the limited domain of regular façades to highlight the importance of the windows in an architectural setting with one single image to create an impressive result of a building façade while depth is manually assigned. Although, this technique is good for modeling regular buildings, it is limited to simple repetitive façades and cannot be applicable to street-view data as in Figure 1. Oh et al. [2001] presented an interactive system to create models from a single image. They also manually assigned the depth based on a painting metaphor. van den Hengel et al. [2007] used a sketching approach in one (or more) image. Although this method is quite general, it is also difficult to scale up for large-scale reconstruction due to the heavy manual interaction. There are also a few manual modeling solutions on the market, such as Adobe Canoma, RealViz ImageModeler, Eos Systems PhotoModeler and The Pixel Farm PFTrack, which all require tedious manual model parameterizations and point correspondences.

Vision-based methods. Vision-based methods automatically reconstruct urban scenes from images. The typical examples are the work in [Snavely et al. 2006; Goesele et al. 2007] , [Cornelis et al. 2008] and the dedicated urban modeling work pursued by University of North Carolina at Chapel Hill and University of Kentucky (UNC/UK) [Pollefeys et al. 2007] that resulted in meshes on dense stereo reconstruction. Proper modeling with man-made structural constraints from reconstructed point clouds and stereo data has not yet been addressed. Werner and Zisserman [2002] used line segments to reconstruct buildings. Dick et al. [2004] developed 3D architectural modeling from short image sequences. The approach is Bayesian and model based, but it relies on many specific architectural rules and model parameters. Lukas et al. [2006; 2008] developed a complete system of urban scene modeling based on aerial images. The result looks good from the top view, but not from the ground level. Our approach is therefore complementary to their system such that the street level details are added. Früh and Zakhor [2003] also used a combination of aerial imagery, ground color and LIDAR scans to construct models of façades. However, like stereo methods, it suffers from the lack of representation for the styles in man-made architectures. Agarwala et al. [2006] composed panoramas of roughly planar scenes without producing 3D models.

3 Overview

Our approach is schematized in Figure 2.

SFM From the captured sequence of overlapping images, we first automatically compute the structure from motion to obtain a set of semi-dense 3D points and all camera positions. We then register the reconstruction with an existing approximate model of the buildings (often recovered from the real images) using GPS data if provided or manually if geo-registration information is not available.

Façade initialization We start a building façade as a flat rectangular plane or a developable surface that is obtained either automatically from the geo-registered approximate building model or we manually mark up a line segment or a curve on the projected 3D points onto the ground plane. The texture image of the flat façade is computed from the multiple visible images of the façade. The detection of occluding objects in the texture composition is possible thanks to the multiple images with parallaxes.

Façade decomposition A façade is then systematically decomposed into a partition of rectangular patches based on the horizontal and vertical lines detected in the texture image. The decomposition is carried out top-down by a recursive subdivision and followed by a bottom-up merging, with detection of the architectural bilateral symmetry and repetitive patterns. The partition is finally structured into a Directed Acyclic Graph of rectilinear elementary patches. We also allow the user to edit the partition by simply adding and removing horizontal and vertical lines.

Façade augmentation Each subdivided patch of the flat façade is augmented with the depth obtained from the MAP estimation of the Markov Random Field with the data cost defined by the 3D points from the structure from motion.

Façade completion The final façade geometry is automatically retextured from all input images.

Our main technical contribution is the introduction of a systematic decomposition schema of the façade that is structured into a Direct Acyclic Graph and implemented as a top-down recursive subdivision and bottom-up merging. This representation strongly embeds the architectural prior of the façades and buildings into different stages of modeling. The proposed optimization for façade depth is also unique in that it operates in the façade surface and in the super-pixel level of a whole subdivision region.

4 Image Collection

Image capturing We use a camera that usually faces orthogonal to the building façade and moves laterally along the streets. The camera should preferably be held straight and the neighboring two views should have sufficient overlapping to make the feature correspondences computable. The density and the accuracy of the reconstructed points vary, depending on the distance between the camera and the objects, and the distance between the neighboring viewing positions.

Structure from motion We first compute point correspondences and structure from motion for a given sequence of images. There are standard computer vision techniques for structure from motion [Hartley and Zisserman 2004]. We use the approach described in [Lhuillier and Quan 2005] to compute the camera poses and a semi-dense set of 3D point clouds in space. This technique is used because it has been shown to be robust and capable of providing sufficient point clouds for object modeling purposes.

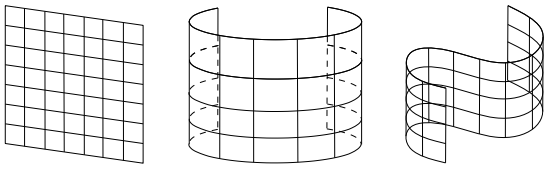


Figure 3: *A simple façade can be initialized from a flat rectangle (a), a cylindrical portion (b) or a developable surface (c).*

5 Façade Initialization

In this paper, we consider that a façade has a dominant planar structure. Therefore, a façade is a flat plane with a depth field on the plane. We also expect and assume that the depth variation within a simple façade is moderate. A real building façade having complex geometry and topology could therefore be broken down into multiple simple façades. A building is merely a collection of façades, and a street is a collection of buildings. The dominant plane of the majority of the façades is flat, but it can be curved sometimes as well. We also consider the dominant surface structure to be any cylinder portion or any developable surface that can be swept by a straight line as illustrated in Figure 3. To ease the description, but without loss of generality, we use a flat façade in the remainder of the paper. For the developable surface, the same methods as for flat façades in all steps are used, with trivial surface parameterizations. Some cylindrical façade examples are given in the experiments.

Algorithm 1 Photo Consistency Check For Occlusion Detection

Require: A set of N image patches $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}$ corresponding to the projections $\{\mathbf{x}_i\}$ of the 3D point $\mathbf{X}$.
Require: $\eta \in [0, 1]$ to indicate when two patches are similar.
for all $\mathbf{p}_i \in \mathcal{P}$ **do**
  $s_i \leftarrow 0$ ▷ Accumulated similarity for $\mathbf{p}_i$
  for all $\mathbf{p}_j \in \mathcal{P}$ **do**
    $s_{ij} \leftarrow \mathrm{NCC}(\mathbf{p}_i, \mathbf{p}_j)$
    if $s_{ij} > \eta$ **then** $s_i \leftarrow s_i + s_{ij}$
    end if
  end for
end for
$\widehat{n} \leftarrow \arg\max_i s_i$ ▷ $\widehat{n}$ is the patch with best support
$\mathcal{V} \leftarrow \emptyset$ ▷ $\mathcal{V}$ is the index set with visible projection
$\mathcal{O} \leftarrow \emptyset$ ▷ $\mathcal{V}$ is the index set with occluded projection
for all $\mathbf{p}_i \in \mathcal{P}$ **do**
  if $s_{i\widehat{n}} > \eta$ **then** $\mathcal{V} \leftarrow \mathcal{V} \cup \{i\}$
  else $\mathcal{O} \leftarrow \mathcal{O} \cup \{i\}$
  end if
end for
return $\mathcal{V}$ and $\mathcal{O}$

5.1 Initial Flat Rectangle

The reference system of the 3D reconstruction can be geo-registered using GPS data of the camera if available or using an interactive technique. Illustrated in Figure 2, the façade modeling process can begin with an existing approximate model of the buildings often reconstructed from areal images, such as publicly available from Google Earth and Microsoft Virtual Earth. Alternatively, if no such approximate model exists, a simple manual process in the current implementation is used to segment the façades, based on the projections of the 3D points to the ground floor. We draw a line segment or a curve on the ground to mark up a façade plane as a flat rectangle or a developable surface portion. The plane or surface position is automatically fitted to the 3D points or manually adjusted if necessary.

5.2 Texture Composition

The geometry of the façade is initialized as a flat rectangle. Usually, a façade is too big to be entirely observable in one input image. We first compose a texture image for the entire rectangle of the façade from the input images. This process is different from image mosaic, as the images have parallax, which is helpful for removing the undesired occluding objects such as pedestrians, cars, trees, telegraph poles and trash cans, that lies in front of the target façade. Furthermore, the façade plane position is known, compared with an unknown spatial position in stereo algorithms. Hence, the photo consistency constraint is more efficient and robust for occluding object removal, with a better texture image than a pure mosaic.

Multi-view occlusion removal As in many multiple view stereo methods, photo consistency is defined as follows. Consider a 3D point $\mathbf{X} = (x, y, z, 1)'$ with color $\mathbf{c}$. If it has a projection, $\mathbf{x}_i = (u_i, v_i, 1)' = \mathbf{P}_i\mathbf{X}$ in the i-th camera $\mathbf{P}_i$, under the Lambertian surface assumption, the projection $\mathbf{x}_i$ should also have the same color, $\mathbf{c}$. However, if the point is occluded by some other objects in this camera, the color of the projection is usually not the same as $\mathbf{c}$. Note that $\mathbf{c}$ is unknown. Assuming that point $\mathbf{X}$ is visible from multiple cameras, $\mathcal{I} = \{\mathbf{P}_i\}$, and occluded by some objects in the other cameras, $\mathcal{I}' = \{\mathbf{P}_j\}$, then the color, $\mathbf{c}_i$, of the projections in $\mathcal{I}$ should be the same as $\mathbf{c}$, while it may be different from the color, $\mathbf{c}_j$, of projections in $\mathcal{I}'$. Now, given a set of projection colors, $\{\mathbf{c}_k\}$, the task is to identify a set, $\mathcal{O}$, of the oc-

(a) Indicate (b) Remove (c) Inpaint (d) Guide (e) Result

Figure 4: *Interactive texture refinement: (a) drawn strokes on the object to indicate removal. (b) the object is removed. (c) automatically inpainting. (d) some green lines drawn to guide the structure. (e) better result achieved with the guide lines.*

cluded cameras. In most situations, we can assume that point $\mathbf{X}$ is visible from most of the cameras. Under this assumption, we have $\widehat{\mathbf{c}} \approx \text{median}_k \{\mathbf{c}_k\}$. Given the estimated color of the 3D point $\widehat{\mathbf{c}}$, it is now very easy to identify the occluded set, $\mathcal{O}$, according to their distances with $\widehat{\mathbf{c}}$. To improve the robustness, instead of a single color, the image patches centered at the projections are used, and patch similarity, normalized cross correlation (NCC), is used as a metric. The details are presented in Algorithm 1. In this way, with the assumption that the façade is almost planar, each pixel of the reference texture corresponds to a point that lies on the flat façade. Hence, for each pixel, we can identify whether it is occluded in a particular camera. Now, for a given planar façade in space, all visible images are first sorted according to the fronto-parallelism of the images with respect to the given façade. An image is said to be more fronto-parallel if the projected surface of the façade in the image is larger. The reference image is first warped from the most fronto-parallel image, then from the lesser ones according to the visibility of the point.

Inpainting In each step, due to existence of occluding objects, some regions of the reference texture image may still be left empty. In a later step, if an empty region is not occluded and visible from the new camera, the region is filled. In this way of a multi-view inpainting, the occluded region is filled from each single camera. At the end of the process, if some regions are still empty, a normal image inpainting technique is used to fill it either automatically [Criminisi et al. 2003] or interactively as described in Section 5.3. Since we have adjusted the cameras according to the image correspondences during bundle adjustment of structure from motion, this simple mosaic without explicit blending can already produce very visually pleasing results.

5.3 Interactive Refinement

As shown in Figure 4, if the automatic texture composition result is not satisfactory, a two-step interactive user interface is provided for refinement. In the first step, the user can draw strokes to indicate which object or part of the texture is undesirable as in Figure 4(a). The corresponding region is automatically extracted based on the input strokes as in Figure 4(b) using the method in [Li et al. 2004]. The removal operation can be interpreted as that the most fronto-parallel and photo-consistent texture selection, from the result of Algorithm 1, is not what the user wants. For each pixel, $\widehat{n}$ from Line 9 of Algorithm 1 and $\mathcal{V}$ should be wrong. Hence, $\mathcal{P}$ is updated to exclude $\mathcal{V}$: $\mathcal{P} \leftarrow \mathcal{O}$. Then, if $\mathcal{P} \neq \emptyset$, Algorithm 1 is run again. Otherwise, image inpainting [Criminisi et al. 2003] is used for automatically inpainting as in Figure 4(c). In the second step, if the automatic texture filling is poor, the user can manually specify important missing structural information by extending a few curves or line segments from the known to the unknown regions as in Figure 4(d). Then, as in [Sun et al. 2005], image patches are synthesized along these user-specified curves in the unknown region using patches selected around the curves in the known region by Loopy Belief Propagation to find the optimal patches. After completing the structural propagation, the remaining unknown regions

(a) Input (b) Structure (c) Weight (d) Subdivide (e) Merge

Figure 5: *Structure preserving subdivision. The hidden structure of the façade is extracted out to form a grid in (b). Such hypotheses are evaluated according to the edge support in (c), and the façade is recursively subdivided into several regions in (d). Since there is not enough support between Regions A, B, C, D, E, F, G, H, they are all merged into one single region M in (e).*

are filled using patch-based texture synthesis as in Figure 4(e).

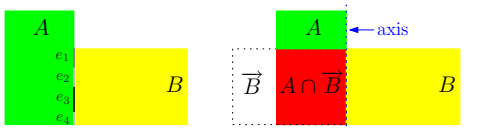
6 Façade Decomposition

By decomposing a façade we try to best describe the faced structure, by segmenting it to a minimal number of elements. The façades that we are considering inherit the natural horizontal and vertical directions by construction. In the first approximation, we may take all visible horizontal and vertical lines to construct an irregular partition of the façade plane into rectangles of various sizes. This partition captures the global rectilinear structure of the façades and buildings and also keeps all discontinuities of the façade substructures. This usually gives an over-segmentation of the image into patches. But this over-segmentation has several advantages. The over-segmenting lines can also be regarded as auxiliary lines that regularize the compositional units of the façades and buildings. Some 'hidden' rectilinear structures of the façade during the construction can also be rediscovered by this over-segmentation process.

6.1 Hidden Structure Discovery

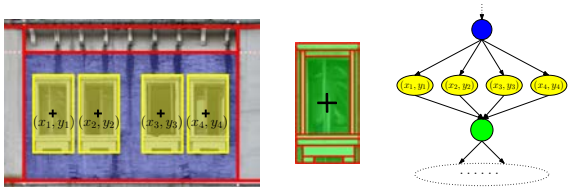
To discover the structure inside the façade, the edge of the reference texture image is first detected [Canny 1986]. With such edge maps, Hough transform [Duda and Hart 1972] is used to recover the lines. To improve the robustness, the direction of the Hough transform is constrained to only horizontal and vertical, which happens in most architectural façades. The detected lines now form a grid to partition the whole reference image, and this grid contains many non-overlapping short line segments by taking intersections of Hough lines as endpoints as in Figure 5(b). These line segments are now the hypothesis to partition the façade. The Hough transformation is good for structure discovery since it can extract the hidden global information from the façade and align line segments to this hidden structure. However, some line segments in the formed grid may not really be a partition boundary between different regions. Hence, the weight, w_e, is defined for each line segment, e, to indicate the likelihood that this line segment is a boundary of two different regions as shown in Figure 5(c). This weight is computed as the number of edge points from the Canny edge map covered by the line segment.

Remark on over-segmented partition It is true that the current partition schema is subject to segmentation parameters. But it is important to note that usually a slightly over-segmented partition is not harmful for the purpose of modeling. A perfect partition certainly eases the regularization of the façade augmentation by depth as presented in the next section. Nevertheless, an imperfect, particularly a slight over-segmented partition, does not affect the modeling results when the 3D points are dense and the optimization works well.



(a) Edge weight support (b) Regional statistics support

Figure 6: *Merging support evaluation.*



(a) Façade (b) DAG

Figure 7: *A DAG for repetitive pattern representation.*

6.2 Recursive Subdivision

Given a region, D, in the texture image, it is divided into two sub rectangular regions, D_1 and D_2, such that $D = D_1 \cup D_2$, by a line segment L with strongest support from the edge points. After D is subdivided into two separate regions, the subdivision procedures continue on the two regions, D_1 and D_2, recursively. The recursive subdivision procedure is stopped if either the target region, D, is too small to be subdivided, or there is not enough support for a division hypothesis, i.e., region D is very smooth.

For a façade, the bilateral symmetry about a vertical axis may not exist for the whole façade, but it exists locally and can be used for more robust subdivision. First, for each region, D, the NCC score, s_D, of the two halves, D_1 and D_2, vertically divided at the center of D is computed. If $s_D > \eta$, region D is considered to have bilateral symmetry. Then, the edge map of D_1 and D_2 are averaged, and subdivision is recursively done on D_1 only. Finally, the subdivision in D_1 is reflected across the axis to become the subdivision of D_2, and merged the two subdivisions into the subdivision of D.

Recursive subdivision is good to preserve boundaries for man-made structural styles. However, it may produce some unnecessary fragments for depth computation and rendering as in Figure 5(d). Hence, as a post-processing, if two neighboring leaf subdivision regions, A and B, has not enough support, s_{AB}, to separate them, they are merged into one region. The support, s_{AB}, to separate two neighbor regions, A and B, is defined to be the strongest weight of all the line segments on the border between A and B: $s_{AB} = \max_e \{w_e\}$. However, the weights of line segments can only offer a local image statistic on the border. To improve the robustness, a dual information region statistic between A and B can be used more globally. As in Figure 6, Since regions A and B may not have the same size, this region statistic similarity is defined as follows: First, an axis is defined on the border between A and B, and region B is mirrored on this axis to have a region, $\overrightarrow{B}$. The overlapped region, $A \cap \overrightarrow{B}$ between A and $\overrightarrow{B}$ is defined to be the pixels from A with locations inside $\overrightarrow{B}$. In a similar way, $\overleftarrow{A} \cap B$ contains the pixels from B with locations inside $\overleftarrow{A}$, and then it is mirrored to become $\overleftarrow{A} \cap \overrightarrow{B}$ according the the same axis. The normalized cross correlation (NCC) between $A \cap \overrightarrow{B}$ and $\overleftarrow{A} \cap \overrightarrow{B}$ is used to define the regional similarity of A and B. In this way, only the symmetric part of A and B is used for region comparison. Therefore, the effect of the other far-away parts of the region is avoided, which will happen if the size of A and B is dramatically different and global statistics, such as the color histogram, are used. Weighted by a parameter, κ, the support, s_{AB}, to separate two neighboring regions, A and B, is now defined as

$$s_{AB} = \max_e \{w_e\} - \kappa \text{NCC}(A \cap \overrightarrow{B}, \overleftarrow{A} \cap \overrightarrow{B}).$$

Note that the representation of the façade is a binary recursive tree before merging and a Directed Acyclic Graph (DAG) after region merging.The DAG representation can innately support the Level of Detail rendering technique. When great details are demanded, the rendering engine can go down the rendering graph to expand all detailed leaves and render them correspondingly. Vice versa, the intermediate node is rendered and all its descendents are pruned at rendering time.

6.3 Repetitive Pattern Representation

The repetitive patterns of a façade locally exist in many façades and most of them are windows. [Müller et al. 2007] used a complicated technique for synchronization of subdivisions between different windows. To save storage space and to ease the synchronization task, in our method, only one subdivision representation for the same type of windows is maintained. Precisely, a window template is first detected by a trained model [Berg et al. 2007] or manually indicated on the texture images. The templates are matched across the reference texture image using NCC as the measurement. If good matches exist, they are aligned to the horizontal or vertical direction by a hierarchical clustering, and the Canny edge maps on these regions are averaged. During the subdivision, each matched region is isolated by shrinking a bounding rectangle on the average edge maps until it is snapped to strong edges, and it is regarded as a whole leaf region. The edges inside these isolated regions should not affect the global structure, and hence these edge points are not used during the global subdivision procedure. Then, as in Figure 7, all the matched leaf regions are linked to the root of a common subdivision DAG for that type of window, by introducing 2D translation nodes for the pivot position. Recursive subdivision is again executed on the average edge maps of all matched regions. To preserve photo realism, the textures in these regions are not shared and only the subdivision DAG and their respective depths are shared. Furthermore, to improve the robustness of the subdivision, the vertical bilateral symmetric is taken as a hard constraint for windows.

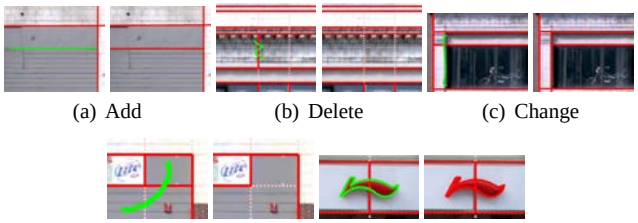
6.4 Interactive Subdivision Refinement

In most situations, the automatic subdivision works satisfactorily. If the user wants to refine the subdivision layout further, three line operations and two region operations are provided. The current automatic subdivision operates on the horizontal and vertical directions for robustness and simplicity. The fifth 'carve' operator allows the user to sketch arbitrarily shaped objects manually, which appear less frequently, to be included in the façade representation.

Add In an existing region, the user can sketch a stroke to indicate the partition as in Figure 8(a). The edge points near the stroke are forced to become salient, and hence the subdivision engine can figure the line segment out and partition the region.

Delete The user can sketch a zigzag stroke to cross out a line segment as in Figure 8(b).

Change The user can first delete the partition line segments and then add a new line segment. Alternatively, the user can directly sketch a stroke. Then, the line segment across by the stroke will be deleted and a new line segment will be constructed accordingly as in Figure 8(c). After the operation, all descendants with the target



(a) Add (b) Delete (c) Change

(d) Group (e) Carve

Figure 8: *Five operations for subdivision refinement: the left figure is the original subdivision layout shown in red and the user sketched stroke shown in green, while the right figure is the resulting subdivision layout.*

region as the root node in the DAG representation will be triggered to be recomputed.

Group The user can draw a stroke to cover several regions to merge them into a single group as in Figure 8(d).

Carve The user can draw line segments or NURBS curves to carve and split the existing subdivision layout as in Figure 8(e). In this way, any shape can be extruded and hence be supported.

7 Façade Augmentation

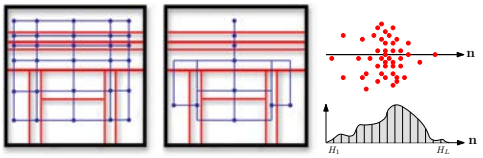
At the previous stage, we obtained a subdivision of the façade plane. In this section, each subdivision region is assigned a depth. If the 3D points are not dense, there might be subdivision regions that could not be assigned a valid depth. These depths can be obtained from the MAP estimation of the Markov Random Field. In traditional stereo methods [Scharstein and Szeliski 2002], a reference image is selected and a disparity or depth value is assigned to each of its pixels. The problem is often formulated as a minimization of the Markov Random Field (MRF) [Geman and Geman 1984] energy functions to provide a clean and computationally tractable formulation. However, a key limitation of these solutions is that the smoothness term imposed by the MRF is viewpoint dependent, in that if a different view were chosen as the reference image, the results could be different. Now with our representation of the façade with the subdivision regions, we could extend the MRF techniques by recovering a surface for each façade, or the depth map on the flat façade plane instead of a depth map on an image plane. Here, we could lay a MRF over the façade surface and define an image and viewpoint independent smoothness constraint.

7.1 Depth Optimization

Suppose the graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V} = \{\mathbf{s}_k\}$ is the set of all sites and $\mathcal{E}$ is the set of all arcs connecting adjacent nodes. The labeling problem is to assign a unique label h_k for each node $\mathbf{s}_k \in \mathcal{V}$. The solution $H = \{h_k\}$ can be obtained by minimizing a Gibbs energy [Geman and Geman 1984]:

$$E(H) = \sum_{\mathbf{s}_k \in \mathcal{V}} E^d_k(h_k) + \lambda \sum_{(\mathbf{s}_k, \mathbf{s}_l) \in \mathcal{E}} E^s_{(k,l)}(h_k, h_l), \quad (1)$$

where $E^d_k(h_k)$ is the data cost (likelihood energy), that encodes the cost when the label of site $\mathbf{s}_k$ is h_k, and $E^s_{(k,l)}(h_k, h_l)$ is the prior energy, denoting the smoothness cost when the labels of adjacent sites, $\mathbf{s}_k$ and $\mathbf{s}_l$, are h_k and h_l, respectively.



(a) Layout 1 (b) Layout 2 (c) Data cost

Figure 9: *Markov Random Field on the façade surface.*

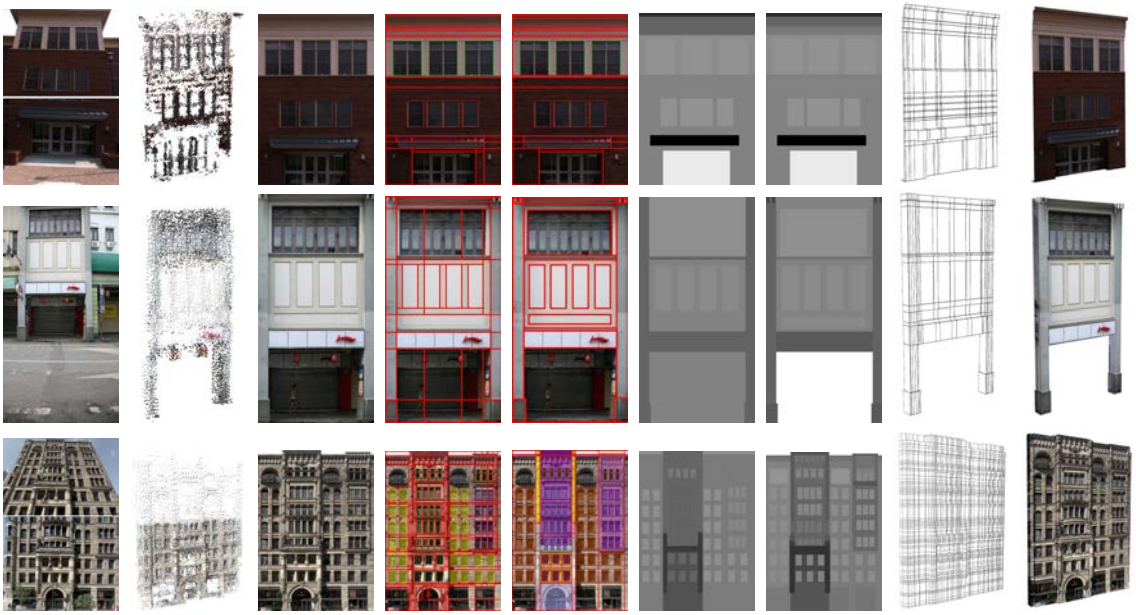
Assume that we have a set of M sample points, $\{\mathbf{s}_k\}_{k=1}^M$, and denote the normal direction of the façade of the initial base surface to be $\mathbf{n}$. The sites of the MRF correspond to height values, $h_1, \ldots, h_M$, measured from the sample points, $\mathbf{s}_1, \ldots, \mathbf{s}_M$, along the normal $\mathbf{n}$. The labels $\{H^k_1, \ldots, H^k_L\}$ are a set of possible height values that variables h_k can take. If the k-th site is assigned label h_k then the relief surface passes through 3D point $\mathbf{s}_k + h_k\mathbf{n}$. Different from [Vogiatzis et al. 2008] where the base surface sample points are uniformly and densely defined, our graph layout is based on the subdivision region and thus is much more efficient for approximation optimization. There are two possible choices to define the MRF graph on the subdivision as shown in Figures 9(a) and 9(b). Layout 1 is good for understanding since it is a regular grid. However, there may be several sites representing the same subdivision region, and this complicates the definition and brings unnecessary scaling-up for optimization. Hence, we prefer to represent each subdivision region with a single site centered at each region as in Layout 2. However, different subdivision regions may have very different areas. This situation is not addressed in the original MRF definition. Hence, the data cost vector, $\mathbf{c}^d_k$, of each site, $\mathbf{s}_k$, is weighted by the area, a_k, in the k-th subdivision region, i.e. $E^d_k = a_k\mathbf{c}^d_k$, and the smoothness cost matrix, $\mathbf{c}^s_{(k,l)}$, is weighted by the length $l_{(k,l)}$ of the border edge between the k-th subdivision region and the l-th subdivision region, i.e. $E^s_{(k,l)} = l_{(k,l)}\mathbf{c}^s_{(k,l)}$.

7.2 Cost Definiton

Traditionally, data cost is defined as the photo consistency between multiple images. Here, we use the photo consistency by means of the point set that we obtain from Structure From Motion. This reverse way of using photo consistency is more robust and is the key reason for achieving the great accuracy of the top performance multi-view stereo method [Furukawa and Ponce 2007]. As shown in Figure 9(c), the 3D points close to the working façade (within 0.8 meter) and with projections inside the subdivision region corresponding to the k-th site, $\mathbf{s}_k$, are projected onto the normal direction, $\mathbf{n}$, to obtain a normalized height histogram, $\boldsymbol{\theta}_k$, with the bins $\{H^k_1, \ldots, H^k_L\}$. The cost vector is now defined as $\mathbf{c}^d_k = \exp\{-\boldsymbol{\theta}_k\}$. Note that if no 3D point exists for a particular region, for example, due to occlusion, a uniform distribution is chosen for $\boldsymbol{\theta}_k$. And the smoothness cost is defined to be

$$\mathbf{c}^s_{(k,l)} = \exp\left\{z_{(k,l)} \left\|(\mathbf{s}_k + h_k\mathbf{n}) - (\mathbf{s}_l + h_l\mathbf{n})\right\|\right\}$$

where $z_{(k,l)}$ is inverse symmetric Kullback-Leibler divergence [Kullback and Leibler 1951] between the normalized color histograms of the k-th region and the l-th region from the reference texture image. This definition penalizes the Euclidean distance between neighboring regions with similar texture and favors minimal area surfaces. Note that $\mathbf{s}_k$ is always placed on the center of the subdivision region, with the depth equal to the peak of the height histogram, $\boldsymbol{\theta}_k$. And $\{H^k_1, \ldots, H^k_L\}$ is adaptively defined to span four standard deviations of the projection heights for the point set. After the costs are all defined on the graph, the energy is minimized by Max-product Loopy Belief Propagation [Weiss and Freeman 2001].



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 10: *Two typical façade examples in the first two rows from different data sets and the most difficult example in the third row: (a) One input view. (b) The 3D points from SFM. (c) The initial textured flat façade. (d) The automatic façade partition (the group of repetitive patterns is color-coded). (e) The user-refined final partition. (f) The re-estimated smoothed façade depth. (g) The user-refined final depth map. (h) The façade geometry. (i) The textured façade model.*

7.3 Interactive Depth Assignment

In most situations, the automatically reconstructed depth is already good enough for visual inspection. For buildings that need more details such as landmarks, our workflow also provides a user interface to facilitate the interactive depth assignment task, all in 2D image space for ease of manipulation.

Transfer from another region If it is not easy to paint the corresponding depth directly, the depth can be transferred from another region by dragging an arrow line to indicate the source and target regions.

Relative depth constraint The relative depth between two regions can also be constrained by dragging a two-circle ended line. The sign symbols in the circles indicate the order, and the radius of the circles, controlled by the + and - key on the keyboard, represent the depth difference. The difference is taken as a hard constraint in the MRF optimization by merging the two nodes in the layout into one and updating the data and smoothness costs accordingly.

8 Façade Completion

Parametrization and texture atlas After the augmentation of the façade with the appropriate depths, each 2D region of the façade partition has evolved from a flat rectangle to a box on the dominant plane of the façade. The parameterization on the façde plane can be used to represent only the front faces of the augmented subdivision regions. The textures of all front faces are stored in one map, the front texture map. The discontinuity between regions due to the difference in depth creates additional side faces: a typical protruding rectangle will have two left and right faces and two up and down faces. The textures of all these side faces are stored in a different side texture map. All textures both for front and side faces are automatically computed from the original registered images using the same method as in Section 5.2.

Re-modeling So far, we approximated each elementary unit of the façade as a cubical box that is sufficient for a majority of the architectural objects on the scale in which we are interested. Obviously, some façades may have elements of different geometries. Each element can be manually re-modeled by using a pre-defined generic cylinder, sphere, and polygonal solid models to replace the given object. The texture is then re-computed automatically from the original images. The columns, arches, and pediments can be modeled this way. Our decomposition approach makes this replacement convenient particularly for a group of elements with automatic texture re-mapping. Figure 11(b) shows a re-modeling example. Again, all textures for re-modeled objects are automatically computed from the original registered images using the same algorithm as in Section 5.2.

9 Experiments

Three representative large-scale data sets captured under different conditions were chosen to show the flexibility of our approach. Video cameras on a vehicle are used for the first data set, a digital camera on a vehicle for the second, and a handheld camera for the third.

For computation of the structure from motion, long sequences were broken down into sub-sequences of about 50 images that are downsampled to the resolution of below 1000×1000. Semi-dense SFM is automatically computed for each subsequence with autocalibration in about 10 minutes with a PC (CPU Intel Core 2 6400 at 2.13GHz and 3GB RAM). The subsequences are merged into a global sequence using one fifth of the reconstructed points from the sub-sequences and the GPS/INS (Inertial Navigation System) data if it is available. To capture tall buildings in full, an additional camera captures views looking upwards in 45 degrees, with little or no overlapping between the viewing fields of the cameras. The cam-

eras are mounted on a rigid rig that can be pre-calibrated, so that viewing positions could be transferable between the cameras if the computation for one camera is difficult.

- **Baity Hill Drive, Chapel Hill.** Shown in Figure 12, these images were captured by two video cameras [Pollefeys et al. 2007] of resolution 1024×768 mounted on a vehicle with a GPS/INS. We sampled a sequences of 308 images from each camera. The resulting clouds of 3D points were geo-registered with the GPS/INS data. The video image quality is mediocre, although the richness of the building texture is excellent for SFM. It took about 20 minutes for the segmentation on the ground. The geometry of the building blocks is rather simple, and it was reconstructed in about one hour.

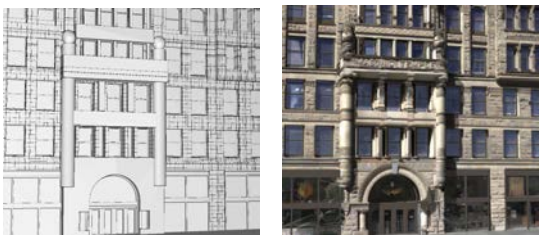
- **Dishifu Road, Canton.** Shown in Figure 1, these images were captured by a handheld Canon 5D camera using a 24mm wide lens with 2912×4368 image resolution. A total of 343 views were captured for one side of the street and 271 views for the opposite side.The arcade is modeled using two façades: a front façade for the top part of the arcade and the columns in the front, and a back façade for its lower part. These two façades are merged together and the invisible ceiling connecting the two parts is manually added. The interactive segmentation on the ground took about one hour and the reconstruction took about two hours for both sides of the street.

- **Hennepin Avenue, Minneapolis.** Shown in Figure 13, these images were captured by a set of cameras mounted on a vehicle equipped with an GPS/INS system. Each camera has a resolution of 1024×1360. The main portion of the Hennepin avenue was covered by a sequence of 130 views using only one of the side-looking cameras. An additional sequence of seven viewing positions taken by an additional side camera pointed 45 degrees up was used for the processing of the structure of the masonic temple to capture the top part of the building. To generate a more homogeneous textured layer from multiple images, the images were white balanced using the diagonal model of illumination change. The Hennepin Avenue in Figure 13 was modeled in about one hour. The Masonic Temple is the most difficult one to model and it took about 10 minutes including re-modeling.

For the rendering results in the video, we assigned different reflective properties for the windows and manually modeled the ground and vegetation. Our approach has been found to be efficient: Most manual post-editing was needed for visually important details near the roof tops of the buildings, where the common coverage of the images is small, and the quality of the recovered point cloud is poor. The re-modeling with generic models for clusters of patches is done only on the Hennepin Avenue example. It is obvious that the accuracy of the camera geometry and the density of reconstructed points are keys to the modeling. GPS/INS data did help to improve the registration of long sequences and avoid the drift associated with the SFM.

Typical façades Some typical façade examples from each data set are shown in Figure 10. An example from the Minneapolis data is also in the flowchart. We show both the before and after editing of the automatic partition, which shows that the majority of the façade partitions can be automatically computed with a over-segmentation followed by minor user adjustments. On average, the automatic computation time is about one minute per façade, then followed by about another minute of manual refinement.depending on the complexity and the desired reconstruction quality.

(a) Two cylindrical façades.



(b) Re-modeling by replacing a cube by a cylinder or a sphere.

Figure 11: *Atypical façades examples: the geometry on the left and the textured model on the right.*

Difficult façade The masonic temple façade in the third row of Figure 10 shows the most difficult case that we encountered, mainly due to the specific capturing conditions. The overlapping of the images for the upper part of the building is small and we reconstruct only few points. The depth map for the upper part is almost constant after optimization. User interaction is more intensive to re-assign the depth for this façade.

Atypical façades Figure 11 shows some special façade examples that are also nicely handled by our approach.

- **Cylindrical façade** An example of two cylindrical façades is illustrated in Figure 11(a). The cylindrical façade with the letters is modeled first; then the cylindrical façade with the windows second; the background façade touched on them is modeled last.

- **Re-modeling** This option was tested in the example of Hennepin Avenue in Figure 13. The re-modeling results with 12 re-modeling objects, shown in Figure 11(b) can be compared with the results obtained without re-modeling shown in the right of Figure 10.

- **Multiple façades** For the topologically complex building façades, we could use multiple façades. The whole Canton arcade street in Figure 1 systematically used two façades, one front and one back, where the back façade uses the front façade in front as the occluders.

10 Discussion and Future Work

We have presented an image-based street-side modeling approach that takes a sequence of overlapping images along the street, and produces complete photo-realistic 3D façade models. Our approach has several limitations for improvement as future work. The automatic depth reconstruction techniques may fail when trying to model highly reflective mirror-like buildings. And the reflectance properties of the models might be automatically recovered from multiple views. Furthermore, non-rectilinear objects might also be automatically detected during partition.

Figure 12: *The modeling of a Chapel Hill street from 616 images (captured by UNC/UK) : two input images on the top left, the recovered model rendered in the bottom row, and two zoomed sections of the recovered model rendered in the middle and on the right of the top row.*

Figure 13: *Modeling of Hennepin Avenue in Minneapolis from 281 images: some input images in the bottom row, the recovered model rendered in the middle row, and three zoomed sections of the recovered model rendered in the top row.*

Acknowledgements

This work was supported by Hong Kong RGC Grants 618908, 619107, 619006, and RGC/NSFC N-HKUST602/05. Ping Tan is supported by Singapore FRC Grant R-263-000-477-112. We acknowledge University of North Carolina at Chapel Hill and University of Kentucky for the data set on Baity Hill Drive in Chapel Hill.

References

AGARWALA, A., AGRAWALA, M., COHEN, M., SALESIN, D., AND SZELISKI, R. 2006. Photographing long scenes with multi-viewpoint panoramas. *ACM Transactions on Graphics(SIGGRAPH) 25*, 3, 853–861.

BERG, A. C., GRABLER, F., AND MALIK, J. 2007. Parsing images of architectural scenes. In *Proceedings of IEEE International Conference on Computer Vision*, 1–8.

CANNY, J. F. 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*, 679–714.

CORNELIS, N., LEIBE, B., CORNELIS, K., AND GOOL, L. V. 2008. 3d urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision 78*, 2–3, 121–141.

CRIMINISI, A., PEREZ, P., AND TOYAMA, K. 2003. Object removal by exemplar-based inpainting. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, vol. 2, 721–728.

DEBEVEC, P., TAYLOR, C., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proceedings of ACM SIGGRAPH*, 11–20.

DICK, A., TORR, P., AND CIPOLLA, R. 2004. Modelling and interpretation of architecture from several images. *International Journal of Computer Vision 2*, 111–134.

DUDA, R. O., AND HART, P. E. 1972. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM 15*, 1, 11–15.

FRÜH, C., AND ZAKHOR, A. 2003. Constructing 3d city models by merging ground-based and airborne views. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, vol. 2, 562–569.

FURUKAWA, Y., AND PONCE, J. 2007. Accurate, dense, and robust multi-view stereopsis. In *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*, 1–8.

GEMAN, S., AND GEMAN, D. 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*, 6, 721–741.

GOESELE, M., SNAVELY, N., CURLESS, B., SEITZ, S. M., AND HOPPE, H. 2007. Multi-view stereo for community photo collections. In *Proceeding of IEEE International Conference in Computer Vision*, 1–8.

HARTLEY, R. I., AND ZISSERMAN, A. 2004. *Multiple View Geometry in Computer Vision*, second ed. Cambridge University Press, ISBN: 0521540518.

KULLBACK, S., AND LEIBLER, R. A. 1951. On information and sufficiency. *Annals of Mathematical Statistics 22*, 79–86.

LHUILLIER, M., AND QUAN, L. 2005. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 3, 418–433.

LI, Y., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2004. Lazy snapping. *ACM Transactions on Graphics 23*, 303–308.

LUKAS, Z., ANDREAS, K., BARBARA, G.-G., AND KONRAD, K. 2006. Towards 3d map generation from digital aerial images. *International Journal of Photogrammetry and Remote Sensing 60*, 413–427.

LUKAS, Z., JOACHIM, B., KONRAD, K., AND HORST, B. 2008. Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In *Proceedings of the European Conference on Computer Vision*.

MÜLLER, P., WONKA, P., HAEGLER, S., ULMER, A., AND GOOL, L. V. 2006. Procedural modeling of buildings. *ACM Transactions on Graphics 3*, 614–623.

MÜLLER, P., ZENG, G., WONKA, P., AND GOOL, L. V. 2007. Image-based procedural modeling of façades. *ACM Transactions on Graphics 26*, 3, 85.

OH, B. M., CHEN, M., DORSEY, J., AND DURAND, F. 2001. Image-based modeling and photo editing. *ACM Transactions on Graphics 1*, 433–442.

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., AND TOWLES, H. 2007. Detailed real-time urban 3d reconstruction from video. *International Journal of Comptuer Vision 78*, 2–3, 143–167.

SCHARSTEIN, D., AND SZELISKI, R. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision 47*, 1/2/3, 7–42.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: Exploring photo collections in 3d. *ACM Transactions on Graphics 3*, 835–846.

SUN, J., YUAN, L., JIA, J., AND SHUM, H.-Y. 2005. Image completion with structure propagation. *ACM Transactions on Graphics 24*, 861–868.

VAN DEN HENGEL, A., DICK, A., THORMÄHLEN, T., WARD, B., AND TORR, P. H. S. 2007. Videotrace: rapid interactive scene modelling from video. *ACM Transactions on Graphics 3*, 86.

VOGIATZIS, G., TORR, P. H. S., SEITZ, S. M., AND CIPOLLA, R. 2008. Reconstructing relief surfaces. *Image and Vision Computing 26*, 3, 397–404.

WEISS, Y., AND FREEMAN, W. T. 2001. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory 47*, 2, 723–735.

WERNER, T., AND ZISSERMAN, A. 2002. New techniques for automated architectural reconstruction from photographs. In *Proceedings of the European Conference on Computer Vision*, vol. 2, 541–555.

WONKA, P., WIMMER, M., SILLION, F., AND RIBARSKY, W. 2003. Instant architecture. *ACM Transactions on Graphics 4*, 669–677.