Large-Scale Urban Modeling by Combining Ground Level Panoramic and Aerial Imagery

Lu Wang, Suya You and Ulrich Neumann
Department of Computer Science
University of Southern California
Los Angeles, CA 90089
{luwang, suyay, uneumann}@graphics.usc.edu

Abstract

We describe an interactive system that models regions of an urban environment, such as a group of tall buildings. Traditional image-based modeling methods often cannot model such large areas due to error accumulation and limited camera field of view. Our approach widens the camera field of view by constructing a 360 degree panorama from ground-level images and uses a high resolution orthorectified aerial image to provide the building footprints. Users draw the building outlines in the aerial image and select a point as the approximate ground camera location. The method automatically extracts roof corners in the ground images and registers the panorama to the aerial image according to geometric constraints. The height of each building is calculated from an estimated camera pose. The resulting textured model of the buildings is constructed of planar surfaces.

1. Introduction

3D scene reconstruction from images is a classic problem in computer vision. While significant progresses have been made, accurate large-scale urban modeling still poses difficult problems that require skill and time to overcome. The presented method focuses on the modeling of a large urban area of multiple tall buildings.

Existing methods for large-scale modeling mostly depend on remote sensing techniques such as stereoscopic aerial images and airborne LIDAR. A good survey is presented in [8]. However, the models generated by these methods lack facade information. In [6, 15], ground-level laser scanners are used together with images to create excellent models but these systems require laser scanners. Our method only requires an off-the-shelf camera and an aerial image to create textured three-dimensional scene models.

There is much prior research related to architectural modeling with images. Some methods are based on self-calibration techniques [2, 7, 11]. Since only image information is used, they often suffer from erroneous feature matchings and some ambiguities in the computation. Fortunately, man-made environment is often highly structured. Parallel and orthogonal relationships between lines and surfaces are abundant. In [4, 5], these geometric constraints are input by users. In [17], vanishing points corresponding to orthogonal directions are automatically detected. Using such prior knowledge about the scene, these approaches are often more robust. However, most of the aforesaid methods are intended to model a single building or part of a building. Due to error propagation, they are difficult to scale up to model a larger scene, such as a group of tall buildings.

To overcome error propagation, some researchers utilized GPS and compass sensors [16]. These instruments can help provide camera pose. Other researchers realized that an orthorectified aerial image is also very helpful. In [6], the model acquired by ground-level laser scans is adjusted to match an overhead aerial image. From building outlines in an orthorectified aerial image, we can identify the footprints(up to a common scale) of the buildings, including such information as the width of a building and the distance between buildings. These measures are used as constraints on the model to be constructed. Current remote sensing technologies provide high resolution aerial images that are inexpensive and widely available. Internet downloads [1] of orthorectified aerial images are available for all major US cities with resolutions as high as one-foot per pixel. Such images are provided from the United States Geographic Survey. One aerial image is shown in Fig. 1(a).

Another inherent problem in large-scale modeling is the limitation of the camera field of view. Many methods require that some minimum numbers of features providing geometric constraints be contained in each image. This can

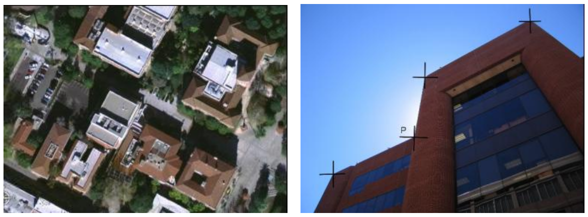
[1] Such as http://earth.google.com and http://terraserver.microsoft.com

be difficult to achieve when buildings are large and streets are narrow. Panorama is hence used by some researchers in architectural modeling[14, 16]. By stitching multiple images taken with a rotating camera, a panorama can have a 360 degree field of view providing sufficient features for modeling wide-area scene.

Our method combines ground-level panoramas and an orthorectified aerial image. The aerial image provides us the buildings footprints. The panorama enables us to estimate the building heights. In [12], D.P.Robertson and R. Cipolla used a map in a similar way as we use the orthorectified aerial image. The difference is that they estimated the camera pose for each ground view separately whereas we combine the constraints from all the ground images taken at the same viewpoint so that the estimated camera pose is more accurate. More importantly, we automate the approach to register the panorama to the aerial image whereas in [12] the registration between ground views and the map is done manually.

In our system, users need only draw the building outlines in the aerial image and click one point to indicate the approximate camera location (not necessary accurate). The system then automatically creates a photorealistic 3D model of a group of buildings and computes the optimal camera pose. Since the model and the camera are registered to a global world frame, it is quite feasible to extend the 3D model by adding more panoramas.

The rest of the paper is organized as follows: Section 2 presents an overview of our approach. Section 3 briefly describes how to create panoramas. In section 4, we show the method to extract roof corners in the ground-level images. The algorithm to automatically register the panorama to the aerial image is described in section 5. Then a bundle adjustment process to refine the initial camera parameter estimates is described in section 6. Section 7 explains the procedure of actually creating the 3D model, and some experimental results are provided in section 8. The paper is closed by a conclusion in section 9.



(a) (b)

Figure 1. (a)An aerial image. (b)Roof corners of a tall building are on the sky boundary.

2. Overview of our approach

Our approach is based on three reasonable assumptions about the building outline in an orthorectified aerial image. (1)Each corner on the outline corresponds to a 3D roof corner. (2)Each line segment of the outline is the orthographic projection of a 3D horizontal line joining two roof corners. (3)There is a vertical wall plane passing each such line segment.

The first two assumptions are valid for most buildings and a few exceptions will not impair our method. These assumptions imply that there is a mapping between the features (corners and line segments) in the aerial image and the features in the ground-level images. This mapping relationship helps us estimate the camera pose. The last assumption is only used for coarse model generation. Approaches to capture more detailed structure by combining images from multiple panoramas will not be covered in this paper. The outline of our approach is as follows:

1. Construct a panorama from a set of ground-level images taken by a rotating camera. This process allows for the estimation of the camera focal length and the relative rotation between the images.

2. Extract roof corners in the ground-level images.

3. Find the correspondences between the roof corners in the ground-level images and the corners on the building outlines in the aerial image. Simultaneously, we obtain an initial estimation of the ground-level camera pose relative to the world frame.

4. Refine the estimated camera parameters through a bundle adjustment process.

5. Compute the height of the buildings and create the textured 3D model.

3 Panorama construction

The images taken by a camera rotating around a single view point are related by a homography between each other [7]. We number these images and take the first as the reference image and regard its camera frame as the reference camera. We call the camera frame of image i as camera i. Suppose $\mathbf{p}$ and $\mathbf{p}'$ are the homogeneous coordinates of two corresponding points in image i and image j respectively, we have:

$$\mathbf{p} = \mathbf{H}_{ji}\mathbf{p}' \tag{1}$$

and

$$\mathbf{H}_{ji} = \mathbf{K}\left(\mathbf{R}_i\right)^{\mathrm{T}}\mathbf{R}_j\mathbf{K}^{-1}, \tag{2}$$

where

$$\mathbf{K} = \begin{bmatrix} f & 0 & cx \\ 0 & f & cy \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

is the calibration matrix of the camera and $\mathbf{R}_i$ is the rotation matrix from camera i to the reference camera. Our goal at this step is to compute the focal length f and the rotation $\mathbf{R}_i$ from each camera i to the reference camera. This can be achieved in the process of constructing a panorama. The technology of panorama construction has matured and we apply the same method as in [3]. In practice, the camera can be rotated on a tripod but our experiments show that we obtain satisfying results even when the camera is held by hand.

4 Feature extraction in ground- level images

Building roof corners and edge segments are the only salient features that can be seen in both the aerial image and the ground-level images. The next step of our approach extracts roof corners from the ground-level images. Unlike general corner detection, in the case of tall buildings, this detection is simplified since most of the roof corners are on the boundary between the sky region and the building area. An example is shown in Fig. 1(b). In these cases, sky regions can be robustly segmented. Therefore, feature extraction has two steps: sky region detection and roof corner extraction from the sky boundary.

4.1 Sky detection

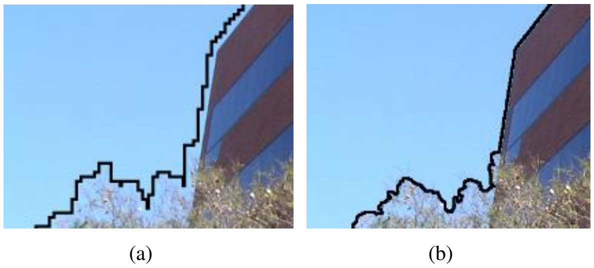
The sky region has several salient characteristics. (1)It is a homogeneous area (we assume an absence of small clouds in the image). (2)Its color has strong blue component.(The dominant color may be sampled for specific sky conditions.) (3)The sky always extends to the upper image border. According to these observations, our sky detection method has the following steps:

First, the image is divided into square patches (for 1600×1200 images, we choose 6×6 as the square size). For the pixels within each patch, compute the variance v_r, v_g, v_b and the mean m_r, m_g, m_b of the three color components. The patches with all v_r, v_g, v_b values below an upper bound (such as 5) and m_b above a lower bound (such as 100) are kept as candidate sky patches.

Next, the connected candidate patches are merged if the difference between their means of each color component is smaller than a threshold (such as 5). In this way, we group candidate patches into several contiguous regions among which the ones adjacent to the upper image border are kept as the sky regions.

Since the sky region obtained so far is composed of square patches, its boundary is not the accurate sky boundary (see Fig. 2(a)). Hence, in the third step we apply graph cut [13] to compute a tighter segmentation in the area surrounding the initial boundary. An example of the final boundary is shown in Fig. 2(b).

Our method is a relatively simple method to segment the sky. More sophisticated approaches [9]are available to improve the robustness.


(a)

(b)

Figure 2. (a)Initial sky boundary. (b)Final sky boundary

4.2 Roof corner extraction

The boundary of the sky region is composed of the outline of buildings and the edges of vegetation. The roof corners are the junctions of the line segments on the building outline. To detect roof corners, we first distinguish building outline from the rest of the sky boundary. This is possible since the building outline is composed of straight-line segments whereas the edges of vegetation are generally irregular. Therefore, our approach is to find a polygonal approximation of the sky boundary and we treat the long line segments of the polygon as belonging to the building outline. The algorithm to detect roof corners proceeds as follows.

The method of [10] is applied to find the optimal polygonal approximation of the sky boundary. Line segments longer than a threshold (50 pixels for 1600×1200 images)are treated as building edges. Long edge segments of structures such as power poles and long protuberances on the roof are detected (and rejected) by finding pairs of line segments with their maximum perpendicular distance within a threshold. Next, we merge adjacent line segments that can be approximated by a single line. Among the remaining line segments, each one is regarded as part of the building outline and their endpoints are kept as roof corners.

The same roof corners may be visible in multiple images. To identify these correspondences, corners in each ground-level image are transferred to its adjacent images by using Eq. 1 and the spatially approximate corners are treated as the projections of the same 3D roof corner.

The output of this stage is a set of corners in the ground-level images. Each corner corresponds to a unique 3D point and is saved together with the identity of its supporting image. In Fig. 5(c)- 5(d), the detected roof corners are marked.

5 Corner matching between the ground images and the aerial image

We note that many of the detected corners in the ground-level images are not roof corners; they arise from occlusions or structures on the roof. In addition, some real roof corners may not be detected since they fall inside the building areas. Furthermore, unlike traditional feature matching, color information cannot help here. Despite these difficulties, automatic correspondence detection is still possible by exploiting camera geometry.

5.1 Coordinate system and parameterization of the reference camera pose

We set up the world frame in the following way. Imagine the image plane of an orthorectified aerial image as the ground plane. The world origin is set to right above the origin of the aerial image. The height of the world origin above the ground plane is the same as the height of the camera center (approximately the height of a human). The x and y directions of the world frame are parallel to the x and y axis of the aerial image. Therefore, the world frame differs from the reference camera frame by a 3D rotation and a 2D translation.

Like many other architectural modeling approaches, we first detect the vanishing point of the 3D lines perpendicular to the ground plane in the reference image, and call it the vertical vanishing point. In our case, this point is detected efficiently because many of the line segments on the building outline detected in the previous stage are projections of vertical 3D lines. We use the approach in [1] to detect the vertical vanishing point from these line segments.

Suppose the rotation from the world frame to the reference camera frame is $R_w = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$, where $\mathbf{r}_i$ is the ith column of R_w. It is well known [7]that given the vanishing point $\mathbf{v}$ of the world z axis, we can compute $\mathbf{r}_3$:

$$\mathbf{r}_3 = \mathbf{K}^{-1}\mathbf{v}. \tag{4}$$

and $\mathbf{r}_3$ is the direction of the world z axis in the reference camera frame. So once we get the vertical vanishing point, there is only 1 DOF left to determine R_w: the rotation of the world frame around its z axis. Let $\mathbf{r}_3 = (r_{13}, r_{23}, r_{33})^{\mathrm{T}}$. It is easy to test that

$$\mathbf{R}_w^0 = \frac{1}{\lambda}\begin{bmatrix} r_{13}r_{33} & -r_{23} & \lambda r_{13} \\ r_{23}r_{33} & r_{13} & \lambda r_{23} \\ -r_{13}^2 - r_{23}^2 & 0 & \lambda r_{33} \end{bmatrix} \tag{5}$$

is a possible instance of R^w, where $\lambda = \sqrt{r_{13}^2 + r_{23}^2}$ (if $r_{13}^2 + r_{23}^2 = 0$, we can choose the identity matrix as $\mathbf{R}_w^0$). Hence, we can express the family of R_w as:

$$\mathbf{R}_w(\alpha) = \mathbf{R}_w^0 \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6}$$

Given α and the location of the camera center in the aerial image $\mathbf{o} = (o_x, o_y)^{\mathrm{T}}$ (the orthographic projection of the camera center on the ground plane), the pose of the reference camera in the world frame is fully determined. In other words, it can be parameterized by $\theta = \{\alpha, o_x, o_y\}$.

5.2 Pose estimation of the reference camera

When a user draws the building outline in the aerial image, the system detects the corners on this outline. These corners correspond to 3D roof corners. From a specific camera location, many of them will not be seen in the ground-level images since they are occluded. Suppose we are given the reference camera pose θ. We check whether the line segment connecting each corner with the camera location in the aerial image crosses any line segment of the building outline. If it does, then we simply regard it as an invisible corner otherwise it is visible. We collect all the visible corners at the given reference camera pose and denote them as $C = \{\mathbf{c}_i, i = 1, \ldots, n\}$. Let $P = \{\mathbf{p}_j, j = 1, \ldots, m\}$ be the set of corners detected in the ground-level images.

Assume the corresponding corner of $\mathbf{c}_i$ is $\mathbf{p}_j$ in ground-level image k. We note that although we do not know the height of the 3D roof corner corresponding to $\mathbf{c}_i$, we know it is on the 3D line perpendicular to the ground plane at $\mathbf{c}_i$. The ray from the camera center $\mathbf{O}$ to the image point $\mathbf{p}_j$ should intersect with this 3D line. This is shown in Fig. 3. Assume $\mathbf{p}_j'$ is the orthographic projection of $\mathbf{p}_j$ onto the ground plane and $\mathbf{o}$ is the camera location in the aerial image. We call the angle between op_j' and oc_i in the aerial image as the angle difference between $\mathbf{c}_i$ and $\mathbf{p}_j$ and denote it as Δ_{ij}. For a pair of correct corner matches, $\Delta_{ij} = 0$. Given the reference camera pose θ, this angle is easily computed. The direction of the ray $\mathbf{O}p_j$ in the world frame is [7]

$$\mathbf{d} = \mathbf{R}_w^{\mathrm{T}}\mathbf{R}_k\mathbf{K}^{-1}\mathbf{p}_j \tag{7}$$

and

$$\Delta_{ij}(\theta) = \arccos\left(\frac{\overrightarrow{op_j'} \cdot \overrightarrow{oc_i}}{|\overrightarrow{op_j'}||\overrightarrow{oc_i}|}\right), \tag{8}$$

where $\overrightarrow{op_j'} = (d_x, d_y)^{\mathrm{T}}$ and $\overrightarrow{oc_i} = \mathbf{c}_i - \mathbf{o}$.

Therefore, given θ, a method to find the corresponding corner for $\mathbf{c}_i$ is to find the corner $\mathbf{p}_j$ in the ground-level images which has the minimum angle difference with $\mathbf{c}_i$. In

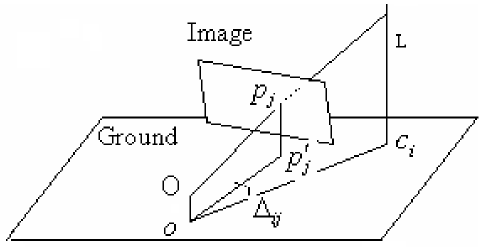


Figure 3. The ray from the camera center o to the image point p_j should intersect with the 3D line L if p_j is the corresponding corner of c_i. o' and p_j' are the projection of o and p_j onto the ground plane.

other words, the correct camera pose θ should minimize the angle differences for all pairs of real corner matches. We call a pair of corner matches with $\Delta_{ij}(\theta)$ smaller than a threshold δ (5 degree in our experiment) as a candidate corner match under θ. In a statistical sense, the correct estimation of the reference camera pose is the one that generates the largest number of candidate corner matches and minimizes the sum of the angle differences of all these matches. For each corner $\mathbf{c}_i$ in the aerial image, we denote

$$\mathrm{D}_i(\theta) = \begin{cases} \min\limits_{p_j \in P} (\Delta_{ij}(\theta)) & \text{if } \min\limits_{p_j \in P} (\Delta_{ij}(\theta)) < \delta; \\ \delta & \text{else.} \end{cases} \tag{9}$$

Then the reference camera pose is estimated as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \left(\sum_{c_i \in C} \mathrm{D}_i(\theta) \right). \tag{10}$$

To solve Eq. (10), we simply do an exhaustive search of $\theta = \{\alpha, o_x, o_y\}$. In our experiments, α ranges from 0 to 360 degrees with a search step of 0.5 degrees. The search range of the camera location is a window centered at the location selected by the user in the aerial image. The window size is 200 × 200(corresponds to a 60 × 60m^2 area on the ground) and the search step is 2-pixels for both o_x and o_y.

5.3 Finding corner matches

Once $\hat{\theta}$ is found, a method to detect the corresponding corner for each $\mathbf{c}_i \in C$ is to find the corner $\mathbf{p}_j \in P$ with the minimum angle difference with $\mathbf{c}_i$. However, this approach may fail when a "noise" corner is very close to the correct one. As shown in Fig. 5(d), A is the correct corner corresponding to a corner $\mathbf{c}$ in the aerial image, but B may have the smallest angle difference with $\mathbf{c}$ due to errors in the estimated camera parameters (K and R_j). Fortunately, we have another cue to help us select the correct corner in this case. According to the second assumption we made in section 2, if two corners in the aerial image are on the same line segment, their corresponding corners are also connected by an obvious boundary in the ground-level images. Hence, if we know C is the corresponding corner of a corner $\mathbf{c}'$ in the aerial image that is connected with $\mathbf{c}$, then we know A must be the correct corresponding corner for C since it is connected with C by an obvious boundary whereas B is not.

To utilize this connection constraint, first we judge whether there is an obvious boundary between any two corners in the ground-level images by computing the average intensity gradient along the line segment connecting them. (If they are on different images, we transfer one corner to be in the same image as the other.)

Given the estimated reference camera pose $\hat{\theta}$, the problem of finding the correct corner match assignment is to find the match that satisfies the connection constraints and minimizes the sum of angle differences between all pairs of corner matches. This optimization is achieved by constructing a graph and finding a shortest path.

We sort the set of corners C in the aerial image into a clockwise order based on the angles formed by the positive x direction in the aerial image to the lines connecting them with the camera location (see Fig. 4(a)). For each corner $c_i \in C$, we find all of its candidate corresponding corners in the ground-level images (the corners with an angle difference with c_i smaller than δ) and denote them as $S_i = \{s_i^k, k = 1, 2, \ldots, m_k\}$. If there are no candidate corresponding corners, then we regard this corner as occluded and delete it from C. The angle difference Δ_{ik} between each s_i^k with $\mathbf{c}_i$ is also stored. The union $V = S_0 \cup S_1 \cup \ldots \cup S_n \cup S_0'$ is the set of nodes in the graph. Note that the nodes in $S_0' = \{(s_0^k)', k = 1, 2, \ldots, m_0\}$ represent the same corners as the nodes in S_0. We add these extra nodes because the corners in C form a cycle.

For any two adjacent corners $\mathbf{c}_i$ and $\mathbf{c}_{i+1}$ in C, edges are created between the nodes in S_i and S_{i+1}. An edge is created between any $s_i^j \in S_i$ and $s_{i+1}^k \in S_{i+1}$ as long as they do not represent the same corner in the ground-level images. If $\mathbf{c}_i$ and $\mathbf{c}_{i+1}$ are connected in the aerial image while the corners represented by s_i^j and s_{i+1}^k are not connected by an obvious boundary in the ground-level images, the cost for the edge connecting these two nodes is set to $\Delta_{ij} + \Delta_{(i+1)k} + \delta$. In other cases, the cost of the edge between s_i^j and s_{i+1}^k is $\Delta_{ij} + \Delta_{(i+1)k}$. δ in the edge cost of the former case is also the threshold for judging candidate corner matches. Here it serves as the penalty for breaking the connection constraint. The constructed graph is shown in Fig. 4(b)

Finally, find the shortest paths from s_0^k to $\left(s_0^k\right)'$ in the graph for all $s_0^k \in S_0$ and compute their costs. The standard shortest path algorithms (such as Dijkstra's algorithm) need to be slightly modified to ensure that no multiple nodes representing the same corner in ground-level images appear in the path. Among all the shortest paths, the one with the lowest cost is kept. The corresponding corner of each $c_i \in C$ is the corner in S_i and on this path.

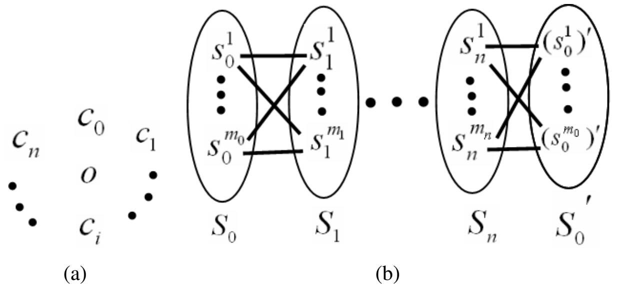


Figure 4. (a)Sort the corners in the aerial image into a clockwise order where o is the camera location. (b)The constructed graph.

6 Bundle adjustment

At this point, all the parameters of the cameras were determined. this section describes a nonlinear optimization to refine these initial estimations so that the geometric constraints can be satisfied as well as possible. The parameters to be optimized include the camera focal length, the rotation $\mathbf{R}_i$from camera i to the reference camera, the rotation $\mathbf{R}_w$ from the world frame to the reference camera frame and the 2D location $\mathbf{o} = (o_x, o_y)^{\mathrm{T}}$ of the camera in the world frame. (The height of the camera remains set to 0 since the world frame is constructed so that the camera center is on its XY plane). There are four sources of constraints to satisfy.

Firstly, during panorama construction, corner matches were detected between adjacent ground-level images. These matches constrain the relative rotations between ground-level camera frames. Suppose corner $\mathbf{p}$ in image i matches corner $\mathbf{p}'$ in image j. This produces the following constraint:

$$\mathbf{p} = \mathbf{K}\mathbf{R}_i^{\mathrm{T}}\mathbf{R}_j\mathbf{K}^{-1}\mathbf{p}'. \tag{11}$$

Secondly, the roof corner correspondences between the aerial image and the ground-level images are detected. These correspondences constrain the reference camera pose in the world frame. Assume that corner $\mathbf{p}$ in ground-level image i corresponds to corner $\mathbf{c} = (c_x, c_y)^{\mathrm{T}}$ in the aerial image. The 2D line connecting $\mathbf{p}$ and the vertical vanishing point $\mathbf{v}$ in image i is the projection of the vertical 3D line L that passes $\mathbf{c}$ on the ground plane. In other words, all the points on L are projected onto the line $\overrightarrow{pv}$ in image i, including the 3D point $\mathbf{q} = (c_x, c_y, 0)^{\mathrm{T}}$ on L. The fact that the projection of $\mathbf{q}$ in image i lies on the line $\overrightarrow{pv}$ provides the constraint:

$$(\mathbf{p} \times \mathbf{v})^{\mathrm{T}}\mathbf{P}_i(c_x, c_y, 0, 1)^{\mathrm{T}} = 0, \tag{12}$$

where $\mathbf{P}_i$ is the projection matrix of camera i

$$\mathbf{P}_i = \mathbf{K}\mathbf{R}_i^{\mathrm{T}}\left[\mathbf{R}_w| - \mathbf{R}_w(o_x, o_y, 0)^{\mathrm{T}}\right]. \tag{13}$$

The vertical vanishing point $\mathbf{v}$ in image i is

$$\mathbf{v} = \mathbf{K}\mathbf{R}_i^{\mathrm{T}}\mathbf{R}_w(0, 0, 1)^{\mathrm{T}} = \mathbf{K}\mathbf{R}_i^{\mathrm{T}}\mathbf{r}_3, \tag{14}$$

where $\mathbf{r}_3$ is the third column of $\mathbf{R}_w$.

Section 2 states the assumption that each line segment on the building outline in the aerial image is the orthographic projection of a horizontal 3D line. This places a constraint on the 3D model to ensure that the two roof corners on this line segment have the same height.This is important in generating a visually pleasing model. Suppose $\mathbf{c}_i = (c_x^i, c_y^i)^{\mathrm{T}}$ and $\mathbf{c}_j = (c_x^j, c_y^j)^{\mathrm{T}}$ are connected corners in the aerial image. $\mathbf{p}_i$ and $\mathbf{p}_j$ are their corresponding corners in the ground-level image i and image j respectively. Transfer corner $\mathbf{p}_i$ and $\mathbf{p}_j$ to the reference ground-level image according to Eq. (1). Denote their corresponding points in the reference image as $\mathbf{p}'_i$ and $\mathbf{p}'_j$. The direction of the horizontal 3D line passing through $\mathbf{c}_i$ and $\mathbf{c}_j$ in the world frame is $\mathbf{d} = (c_x^i - c_x^j, c_y^i - c_y^j, 0)$. The vanishing point v of this direction in the reference ground level image should lie on the line connecting $\mathbf{p}'_i$ and $\mathbf{p}'_j$. This leads to the following constraint:

$$(\mathbf{p}'_i \times \mathbf{p}'_j)^{\mathrm{T}}\mathbf{K}\mathbf{R}_w\mathbf{d} = 0. \tag{15}$$

The last constraint relates to the vertical vanishing point. In section 5.1, in detecting the vertical vanishing point, a set of line segments are found in the ground-level images that are the projection of 3D vertical lines. The fact that the vertical vanishing point is on each of these lines constrains the reference camera pose. Suppose one of these line segments is in ground-level image i. $\mathbf{p}_1$ and $\mathbf{p}_2$ are its two endpoints. Then we have

$$(\mathbf{p}_1 \times \mathbf{p}_2)^{\mathrm{T}}\mathbf{v} = 0, \tag{16}$$

where $\mathbf{v}$ is the vertical vanishing point in image i and it can be computed with Eq. (14).

Given the above constraints, Levenberg-Marquardt is used to do bundle adjustment [7]. Camera radial distortion is also rectified during this process [7]. In addition, since the resolution of the aerial image is limited and the hand-drawn building outline may not be accurate, the building corners in the aerial image are adjusted around their initial user-determined positions. This is done by treating the coordinates of each corner $\mathbf{c}_i$ in the aerial image as unknowns and adding the following constraint:

$$w_i(\mathbf{c}_i - \mathbf{c}'_i) = \mathbf{0}, \tag{17}$$

where w_i is a large weight and $\mathbf{c}'_i$ is the initial coordinates of the corner, as indicated by the user. In our experiment, we choose $w_i = 300$, so that the adjustment for each corner is within 2 pixels.

7 3D model construction

The building outlines in the orthorectified aerial image provide the dimensions of the building footprints in the XY horizontal plane (up to a common scale). With the third assumption in section 2, that there is a vertical wall-plane passing through each line segment on the building outline in the aerial image, we can already build a "2.5D" model of the buildings that is composed of vertical wall-planes with an unknown height. To create a complete 3D model, the height of each building must be computed.

For each corner $\mathbf{c}$ in the aerial image, assume it has a corresponding corner $\mathbf{p}$ in ground-level image i. The direction of the ray from the camera center to image point $\mathbf{p}$ in the world frame is $\mathbf{d} = \mathbf{R}_w^{\mathrm{T}}\mathbf{R}_i\mathbf{K}^{-1}\mathbf{p}$. The height of the roof corner $\mathbf{c}$ is hence the height of the intersection point of this ray with the 3D line passing through $\mathbf{c}$ and vertical to the ground plane. In the case that these two lines do not exactly intersect in the 3D space due to errors, find the 3D point on the ray $\mathbf{d}$ that has the minimum perpendicular distance to the vertical 3D line. Take the height of this 3D point as the height of the corner $\mathbf{c}$. Denote $\mathbf{d} = (d_x, d_y, d_z)^{\mathrm{T}}$. It is easy to prove that this height is

$$h = \frac{d_z d_x(c_x - o_x) + d_z d_y(c_y - o_y)}{d_x^2 + d_y^2}. \tag{18}$$

Note that the height computed above is not the height from the ground plane, but relative to the world origin. Since the world origin is set to the same height as the camera center, the obtained height for all roof corners is less than the height from the ground plane by a common amount (approximately the height of a human).

Our method does not model the rooftops of the buildings since roof details on a tall building cannot be seen from ground level. Therefore we simply add a flat polygon on the top of the building as a roof. Once the 3D model is complete, the known camera pose relative to the model is used to do projective texture mapping. The final textured 3D model is output as a VRML file.

8 Experiment

The prototype system was tested on a variety of scenes. Fig. 5 shows the results of modeling a large environment with two panoramas. Fig. 5(a) shows the building outline provided by user interaction. The stitched panoramas are shown in Fig. 5(b). Each is composed of 20 images. Fig. 5(c) - 5(d) are two of them. The detected roof corners are marked in these images. Note that many of these detected corners are false roof corners. However, the method correctly found the camera poses. The estimated camera locations are shown in Fig. 5(e) as the cyan balls. Fig. 5(e) - 5(h) show several synthetic views of the 3D model.

Two strategies are used to evaluate the results. The first verifies the constructed model dimensions by measuring the actual buildings with an accurate laser range finder. After multiplying by a common scale (1-foot per pixel), the average errors in the x and y dimensions of these buildings are around 0.5 meters. These errors largely depend on the resolution of the aerial image. The average error of the building heights is around 1.0 meter. For example, the actual height of the building in Fig. 5(d) is $22.8m$ while the height computed by our method is $22.0m$.

The second strategy is to use image geo-referencing to verify the accuracy of the geometry model. By projecting the aerial and ground images onto the geometry models, the errors resulting from model reconstruction become visible. This approach reveals the overall qualitative accuracy that includes small-scale building features.

After the user interaction, the modeling process takes about 10 minutes on a 3GHz Pentium 4 computer for one panorama with the image resolution of 1600×1200. Most of the computation is spent on the final bundle adjustment.

9 Conclusion

This paper presents a 3D architectural modeling method that only requires a hand-held digital camera and a high resolution othorectified aerial image. Combining information from both ground-level images and the aerial image, the method not only generates 3D geometry but also provides photo-realistic texture for building facades and overcomes the error accumulation problem in extended environment modeling. The height accuracy of the resulting model is within 5% based on our experiments with actual measurements. The modeling process is automatic once the user provides the building outlines in the aerial image. With further automation of the building segmentation in the aerial image, the user interaction can be eliminated to achieve a fully automated modeling system. This is the direction of our current efforts.

Another limitation to automation is that the automatic roof corner detection method is only suitable for tall buildings, where most of the roof corners appear on the sky boundary in the ground-level images. For short buildings, many of their roof corners may lie inside building areas, requiring the user to manually select roof corners in the ground-level images.However, once selected, the roof corners in ground-level images can still be automatically registered to the corners in the aerial image and the 3D model are created without further manual work.

References

[1] M. E. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. *Computer Vision and Pattern Recognition Conference*, 2:282–289, 2000.

[2] P. A. Beardsley, P. H. S. Torr, and A. Zisserman. 3d model acquisition from extended image sequences. *European Conference on Computer Vision*, 2(2):683–695, 1996.

[3] M. Brown and D. G. Lowe. Recognising panoramas. *International Conference on Computer Vision*, 2:1218–1225, 2003.

[4] R. Cipolla, D. Robertson, and E. Boyer. Photobuilder – 3d models of architectural scenes from uncalibrated images. *International Conference on Multimedia Computing and Systems*, 1:25–31, 1999.

[5] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *SIGGRAPH*, pages 11–20, 1996.

[6] C. Frueh and A. Zakhor. 3d model generation for cities using aerial photographs and ground level laser scans. *Computer Vision and Pattern Recognition Conference*, 2:31–38, 2001.

[7] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. *Cambridge*, 2000.

[8] J. Hu, S. You, and U. Neumann. Approaches to large-scale urban modeling. *IEEE Computer Graphics and Applications*, 23(6):62–69, 2003.

[9] J. Luo and S. P. Etz. A physical model-based approach to detecting sky in photographic images. *IEEE Transactions on Image Processing*, 11(3):201–212, 2002.

[10] M. Marji and P. Siy. Polygonal representation of digital planar curves through dominant point detection–a nonparametric algorithm. *Pattern Recognition*, 37(11):2113–2130, 2004.

[11] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a handheld camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

[12] D. P. Robertsonand and R. Cipolla. Building architectural models from many views using map constraints. *European Conference on Computer Vision*, pages 155–169, 2002.

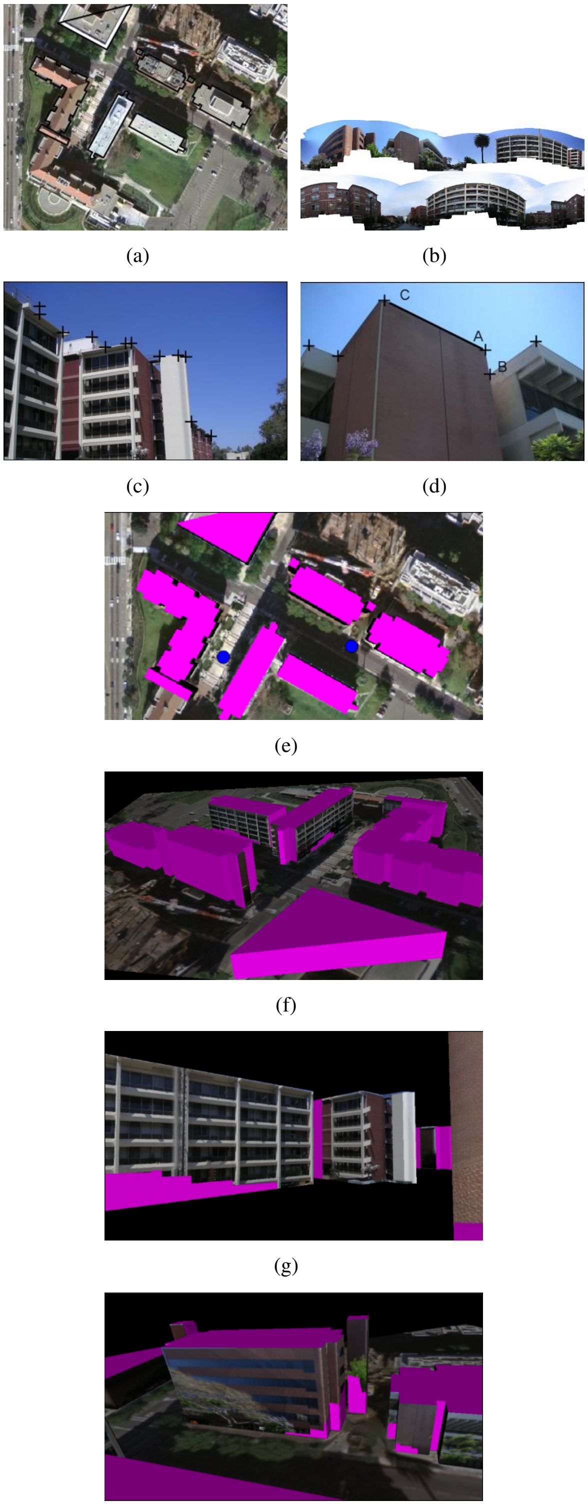
[13] C. Rother, V. Kolmogorov, and A. Blake. “grabcut”: interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.

[14] H. Y. Shum, M. Han, and R. Szeliski. Interactive construction of 3d models from panoramic mosaics. *Computer Vision and Pattern Recognition Conference*, pages 427–433, 1998.

[15] I. Stamos and P. K. Allen. Geometry and texture recovery of scenes of large scale. *Computer Vision and Image Understanding*, 88(2):94–118, 2002.

[16] S. Teller. Automated urban model acquisition: project rationale and status. *Image Understanding Workshop*, pages 455–462, 1998.

[17] T. Werner and A. Zisserman. New techniques for automated architectural reconstruction from photographs. *European conference on computer vision*, 2:541–555, 2002.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 5. (a)Aerial image. (b)Panoramas. (c)-(d)Ground-level images. (e)-(h)Several synthetic views of the 3D model. The two cyan balls in (e) indicate the two camera locations