Chapter 15 Automatic 3D Modeling of Cities with Multimodal Air and Ground Sensors[1]

Avideh Zakhor and Christian Frueh

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley,

[1] Based on “Constructing 3D City Models by Merging Ground-Based and Airborne Views", by Christian Frueh and Avideh Zakhor which appeared in IEEE Transactions on Computer Graphics and Applications, Vol. 23, No. 6, Nov.-Dec. 2003, pp. 52-61. © 2003 IEEE.

15.1 Introduction

Three-dimensional models of urban environments are useful in a variety of applications such as urban planning, training and simulation for disaster scenarios, surveillance applications, and virtual heritage conservation. A standard technique for creating large-scale city models in an automated or semi-automated way is to apply stereo vision techniques on aerial or satellite imagery [9]. In recent years, advances in resolution and accuracy have also rendered airborne laser scanners suitable for generating Digital Surface Models (DSM) and 3D models [2]. Although edge detection can be done more accurately in aerial photos, airborne laser scans are advantageous in that they require no error-prone camera parameter estimation, line or feature detection, or matching. Previous work has attempted to reconstruct polygonal models by using a library of predefined building shapes, or combining the DSM with digital ground plans or aerial images [2, 89-96]. While sub-meter resolution can be achieved using this technique, it is only capable of capturing rooftops, rather than building facades.

There have been several attempts to create models from ground-based view at high level of detail, in order to enable virtual exploration of city environments. While most approaches result in visually pleasing models, they involve an enormous amount of manual work, such as importing the geometry obtained from construction plans, or selecting primitive shapes and correspondence points for image-based modeling, or complex data acquisition. There have also been attempts to acquire close-range data in an automated fashion by using either images [3] or 3D laser scanners [10, 11]. These approaches, however, do not scale to more than a few buildings, since data has to be acquired in a slow stop-and-go fashion.

In this chapter, we review a fast, automated method capable of producing textured 3D models of urban environments for photorealistic walkthroughs, drive-throughs and fly-throughs [6-8]. The resulting models provide both high details at ground level, and complete coverage for bird's eye view. The data flow diagram of our approach is shown in Figure 15.1. The airborne modeling process on the left provides a half-meter resolution model with a bird's-eye view over the entire area, containing terrain profile and building tops. The ground-based modeling process on the right results in a highly detailed model of the building facades [6-8]. The basic idea is to use two different acquisition processes to generate two separate models, namely ground and aerial, to be fused at a later stage. We acquire a close range façade model at the ground level by driving a vehicle equipped with laser scanners and a digital camera under normal traffic conditions on public roads. To our knowledge, our ground based system is the first acquisition system where the data is acquired continuously while the acquisition vehicle is in motion, rather than in a "stop and go" fashion. This combined with automated processing, results in extremely fast model construction schemes for building facades. As for airborne modeling, we create a far range DSM containing complementary roof and terrain shape from airborne laser scans, and after post-processing, texture map it with oblique aerial imagery. The challenge in merging the airborne and ground based models is registration between the two, for which we use the Monte Carlo Localization (MCL) technique. We fuse the two models by first removing redundant parts between the two, and filling the remaining gaps between them. The same MCL process is also used to localize the ground based acquisition vehicle in constructing the 3D façade models.

In the remainder of this chapter, we will describe various steps of the above process in more detail, and show results on 3D models of downtown Berkeley. The outline of this chapter is as follows. In Section 15.2, we describe the basic approach to creating textured surface mesh from airborne laser scans and imagery. In Section 15.3, we go over the ground based data acquisition system and the modeling process, and in Section 15.4, we describe the model merging process. Section 15.5 includes 3D modeling results for downtown Berkeley, and Section 15.6 discusses possible applications of 3D modeling to surveillance problems.

*Figure 15.1**:** Data flow diagram of our modeling approach. Airborne modeling steps are highlighted in green, ground-based modeling steps in yellow, and model fusion steps in white.*

15.2. Creating Textured 3D Airborne Model

In this section, we describe the generation of a DSM from airborne laser scans, its processing and transformation into a surface mesh, and texture-mapping with color aerial imagery. Our basic approach is to use airborne laser scans to model the geometry of rooftops in order to arrive at rectilinear 3D models. We then use oblique airborne imagery acquired by a helicopter to texture map the resulting 3D models. As shown later in Section 15.3.3, we use the DSM resulting from airborne modeling both to localize the ground-based data acquisition vehicle, and to register and merge the ground based model with the airborne one. In contrast to previous approaches, we do not explicitly extract geometric primitives from the

DSM [78,79]. While we use aerial laser scans to create the DSM, it is equally feasible to use a DSM obtained from other sources such as stereo vision or SAR.

15.2.1 Scan Point Re-sampling and DSM Generation

During the acquisition of airborne laser scans with a 2D scanner mounted on board a plane, the unpredictable roll and tilt motion of the plane generally destroys the inherent row-column order of the scans. Thus, the scans may be interpreted as an unstructured set of 3D vertices in space, with the x,y-coordinates specifying the geographical location, and the z coordinate the altitude. In order to further process the scans efficiently, it is advantageous to resample the scan points to a row-column structure, even though this step could potentially reduce the spatial resolution, depending on the grid size. To transfer the scans into a DSM, i.e. a regular array of altitude values, we define a row-column grid in the ground plane, and sort scan points into the grid cells. The density of scan points is not uniform, and hence there are grid cells with no scan points and others with multiple ones. Since the percentage of cells without any scan points and the resolution of the DSM depend on the size of a grid cell, a compromise must be made, leaving few cells without a sample, while maintaining the resolution at an acceptable level.

In our case, the scans have an accuracy of 30 centimeters in the horizontal and vertical directions, and a raw spot spacing of 0.5 meters or less. Both the first and the last pulses of the returning laser light are measured. We have chosen to select a square cell size of 0.5 m × 0.5 m, resulting in about half the cells being occupied. We create the DSM by assigning to each cell the highest z value among its member points, so that overhanging rooftops of buildings are preserved, while points on walls are suppressed. The empty cells are filled using nearest-neighbor interpolation in order to preserve sharp edges. Each grid cell can be interpreted as a vertex, where the x,y location is the cell center, and the z coordinate is the altitude value, or as a pixel at (x,y) with a gray level proportional to z. An example of a re-sampled DSM is shown in Figure 15.2(a).

15.2.2 Processing the DSM

The DSM generated via the above steps contains not only the plain rooftops and terrain shape, but also many other objects such as cars, trees, etc. Roofs, in particular, look “bumpy” due to a large number of smaller objects such as ventilation ducts, antennas, and railings, which are impossible to reconstruct properly at the DSM’s resolution. Furthermore, scan points below overhanging roofs cause ambiguous altitude values, resulting in jittery edges. In order to obtain a more visually pleasing reconstruction of the roofs, we apply several processing steps to the DSM obtained in Section 15.2.1.

The first step is aimed at flattening “bumpy” rooftops. To do this, we first apply a region growing segmentation algorithm to all non-ground pixels based on depth discontinuity between adjacent pixels. Small, isolated regions are replaced with ground level altitude, in order to remove objects such as cars or trees in the DSM. Larger regions are further subdivided into planar sub-regions by means of planar segmentation. Then, small regions and sub-regions are united with larger neighbors by setting their z values to the larger region’s corresponding plane. This procedure is able to remove undesired small objects from the roofs and prevents rooftops from being separated into too many cluttered regions. The resulting processed DSM for Figure 15.2(a) using the above steps is shown in Figure 15.2(b).

The second processing step is intended to straighten jittery edges. We re-segment the DSM into regions, detect the boundary points of each region, and use RANSAC [5] to find line segments that approximate the regions. For the consensus computation, we also consider boundary points of surrounding regions, in order to detect short linear sides of regions, and to align them consistently with surrounding buildings; furthermore, we reward additional bonus consensus if a detected line is parallel or perpendicular to the most dominant line of a region. For each region, we obtain a set of boundary line segments representing the most important edges, which are then smoothed out. For all other boundary parts, where a proper line approximation has not been found, the original DSM is left unchanged. Figure 15.2(c) shows the regions resulting from the above processing steps as applied to Figure 15.2(b), superimposed on top of the corresponding RANSAC lines drawn in white. Compared with Figure 15.2(b), most edges are straightened out.

Figure 15.2: Processing steps for DSM; (a) DSM obtained from scan point re-sampling; (b) DSM after flattening roofs; (c) segments with RANSAC lines in white.

15.2.3. Textured Mesh Generation

Airborne models are commonly generated from LIDAR scans by detecting features such as planar surfaces in the DSM or matching a predefined set of possible rooftop and building shapes [2,78,79]. In other words, they decompose the buildings found in the DSM into polygonal 3D primitives. While the advantage of these model-based approaches is their robust reconstruction of geometry in spite of erroneous scan points and low sample density, they are highly dependent on the shape assumptions that are made. In particular, the results are poor if many non-conventional buildings are present or if buildings are surrounded by trees. Although the resulting models may appear “clean” and precise, the geometry and location of the reconstructed buildings are not necessarily correct if the underlying shape assumptions are invalid.

As we will describe in Section 15.3, in our application, an accurate model of the building facades is readily available from the ground-based acquisition, and as such, we are primarily interested in adding the complementary roof and terrain geometry. Hence, we apply a different strategy to create a model from airborne view, namely transforming the cleaned-up DSM directly into a triangular mesh and reducing the number of triangles by simplification. The advantage of this method is that the mesh generation process can be controlled on a per-pixel level; we exploit this property in the model fusion procedure described in Section 15.4. Additionally, this method has a low processing complexity and is robust: Since no a priori assumptions about the environment are made or pre-defined models are required, it can be applied to buildings with unknown shapes, even in presence of trees. Admittedly, this comes at the expense of a larger number of polygons.

Since the DSM has a regular topology, it can be directly transformed into a structured mesh by connecting each vertex with its neighboring ones. The DSM for a city is large, and the resulting mesh has two triangles per cell, yielding 8 million triangles per square kilometer for the 0.5 m × 0.5 m grid size we have chosen. Since many vertices are coplanar or have low curvature, the number of triangles can be drastically reduced without significant loss of quality. We use the Qslim mesh simplification algorithm [4] to reduce the number of triangles. Empirically, we have found that it is possible to reduce the initial surface mesh to about 100,000 triangles per square kilometer at highest level-of-detail without noticeable loss in quality.

Using aerial images taken with an un-calibrated camera from an unknown pose, we texture-map the reduced mesh in a semi-automatic way. A few correspondence points are manually selected in both the aerial photo and the DSM, taking a few minutes per image. Then, both internal and external camera parameters are automatically computed and the mesh is texture-mapped. Specifically, a location in the DSM corresponds to a 3D vertex in space, and can be projected into an aerial image if the camera parameters are known. We utilize an adaptation of Lowe’s algorithm to minimize the difference between selected correspondence points and computed projections [1]. After the camera parameters are determined, for each geometry triangle, we identify the corresponding texture triangle in an image by projecting the corner vertices. Then, for each mesh triangle the best image for texture-mapping is selected by taking into account resolution, normal vector orientation, and occlusions.

In addition to the above semi-automatic approach, we have developed an automatic method for registering oblique aerial imagery to our airborne 3D models [80]. This is done by matching 2D lines in aerial imagery with projections of 3D lines from the city model, for every camera pose location, and computing the best match with the highest score. We have empirically found techniques such as steepest descent to result in local minima, and hence we have opted to use the above exhaustive search approach. To make exhaustive search computationally feasible, we assume a rough estimate of our initial orientation, i.e. roll, pitch and yaw to be known within a 10 degree range, and our initial position, i.e. x,y, and z to be known within a 20 meter range. These values correspond to the accuracy of a mid-tier GPS/INS system. To achieve reliable pose estimation within the exhaustive search framework, we need to sample the search space at half a degree in roll, a quarter of a degree in pitch and yarw, and 10 meters in x,y, and z. This level of sampling of the parameter space, results in about 20 hours of computation on a 2 GHz Pentium 4 machine per image [80]. This is in sharp contrast with a few minutes per image as achieved by the semi-automatic approach described earlier. We have found the accuracy performance of the two approaches to be close.

Examples of resulting textured, airborne 3D models using the approach described in this section are included in Section 15.5.

15.3. Ground-Based Acquisition and Modeling

In this section, we describe our approach to ground based data acquisition and modeling.

15.3.1. Ground-Based Data Acquisition via Drive-by Scanning

Our mobile ground-based data acquisition system consists of two Sick LMS 2D laser scanners and a digital color camera with a wide-angle lens [6]. The data acquisition is performed in a fast drive-by rather than a stop-and-go fashion, enabling short acquisition times limited only by traffic conditions. As shown in Figure 15.3, our acquisition system is mounted on a rack on top of a truck, enabling us to obtain measurements that are not obstructed by objects such as pedestrians and cars.

Figure 15.3: Acquisition vehicle.

Both 2D scanners face the same side of the street; one is mounted horizontally, the other vertically, as shown in Figure 15.4. The camera is mounted towards the scanners, with its line of sight parallel to the intersection between the orthogonal scanning planes. Laser scanners and camera are synchronized by hardware signals. In our measurement setup, the vertical scanner is used to scan the geometry of the building facades as the vehicle moves, and hence it is crucial to determine the location of the vehicle accurately for each vertical scan. In [6], we have developed algorithms to estimate relative position changes of the vehicle based on matching the horizontal scans, and to estimate the driven path as a concatenation of relative position changes. Since errors in the estimates accumulate, a global correction must be applied. Rather than using a GPS sensor, which is not sufficiently reliable in urban canyons, in [7] we introduce the use of an aerial photo as a 2D global reference in conjunction with MCL. In Section 15.3.3, we extend the application of MCL to a global edge map, and DTM which are both derived from the DSM, in order to determine the vehicle's 6-degree-of-freedom pose in non-planar terrain, and to register the ground-based facade models with respect to the DSM.

Figure 15.4: Ground-based acquisition setup.

15.3.2. Creating Edge Map and DTM

In this section, we describe the methodology for creating the two maps that are needed by MCL to fully localize the ground based acquisition system; the first one is an edge map, which contains the location of height discontinuities in the DSM, and the second one is a Digital Terrain Model (DTM) also obtained from DSM, which contains terrain altitude. In previous work [7], we have applied a Sobel edge detector to a gray scale aerial image in order to find edges which are then used by MCL in conjunction with horizontal scan matches, in order to localize the ground-based data acquisition vehicle. However, since in the process of constructing a complete model we have access to airborne DSM, we opt to use an edge map derived from DSM, rather than one derived from aerial imagery, in order to achieve higher accuracy and faster processing speed [82]. Rather than applying the Sobel edge detector to the DSM though, we construct an edge map by defining a discontinuity detection filter, which marks a pixel if at least one of its eight adjacent pixels is more than a threshold Δz_{edge} below it. This is possible because we are dealing with 3D height maps rather than 2D images. Hence, only the outermost pixels of the taller objects such as building tops are marked, rather than adjacent ground pixels, creating a sharper edge map than a Sobel filter. In fact, the resulting map is a global occupancy grid for building walls. While for aerial photos, shadows of buildings or trees and perspective shift of building tops could potentially cause numerous false edges in the resulting edge map; neither problem exists for the edge map from airborne laser scans.

The DSM contains not only the location of building facades as height discontinuities, but also the altitude of the streets on which the vehicle is driven, and as such, this altitude can be assigned to the z-coordinate of the vehicle in order to create DTM. Nonetheless, it is not possible to directly use the z value of a DSM location, since the LIDAR captures cars and overhanging trees during airborne data acquisition, resulting in z values up to several meters above the actual street level for some locations. To overcome this, for a particular DSM location, we estimate the altitude of the street level by averaging the z-coordinates of available ground pixels within a surrounding window, weighing them with an exponential function decreasing with distance. The result is a smooth, dense DTM as an estimate of the ground level near roads.

Figures 15.5(a) and 15.5(b) show edge map and DTM, respectively, computed from the DSM shown in Figure 15.2(b). MCL technique to be described in the next section utilizes both maps to localize the ground based acquisition vehicle.

Figure 15.5: Map generation for MCL; (a) edge map; (b) DTM. For the white pixels, there is no ground level estimate available.

15.3.3. Model Registration with MCL

MCL is a particle-filtering-based implementation of the probabilistic Markov localization, and was introduced by Thrun et al. [12] for tracking the position of a vehicle in mobile robotics. Our basic approach is to match successive horizontal laser scans in order to arrive at initial estimate the relative motion of the truck on the ground plane [82]. Given a series of relative motion estimates and corresponding horizontal laser scans, we then apply MCL to determine the accurate position of the acquisition vehicle within a global edge map as described in Section 15.3.2 [7,82]. The principle of the correction is to adjust initial vehicle motion estimates so that horizontal scan points from the ground-based data acquisition match the edges in the global edge map. The scan-to-scan matching can only estimate a 3-DOF relative motion, i.e. a 2D translation and rotation in the scanner's coordinate system. If the vehicle is on a slope, the motion estimates are given in a plane at an angle with respect to the global (x,y) plane, and the displacement should in fact be corrected with the cosine of the slope angle. However, since this effect is small, e.g. 0.5 % for a 10%-degree-slope, we can safely neglect it, and use the relative scan-to-scan matching estimates as if the truck's coordinate system were parallel to the global coordinate system. Using MCL with the relative estimates from horizontal scan matching and the edge map from the DSM, we arrive at a series of global pose probability density functions and correction vectors for x, y and yaw. These corrections are then applied to the initial path estimate using horizontal scan matches only, to obtain an accurate localization of the acquisition vehicle. In Section 15.5, we show examples of such corrections on actual data.

Using the DTM as described in Section 15.3.2, an estimate of two more DOF can be obtained: As for the first, the final $z^{(i)}$ coordinate of an intermediate pose P_i in the path is set to DTM level at $(x^{(i)},y^{(i)})$ location; as for the second, the pitch angle representing the slope can be computed as

$$pitch^{(i)} = \arctan\left(\frac{z^{(i)} - z^{(i-1)}}{\sqrt{(x^{(i)} - x^{(i-1)})^2 + (y^{(i)} - y^{(i-1)})^2}}\right),$$

i.e. by using the height difference and the traveled distance between successive positions. Since the resolution of the DSM is only one meter and the ground level is obtained via a smoothing process, the estimated pitch contains only the "low-frequency" components, and not highly dynamic pitch changes e.g.

those caused by pavement holes and bumps. Nevertheless, the obtained pitch is an acceptable estimate, because the size of the truck makes it relatively stable along its long axis.

The last missing DOF, the roll angle, is not estimated using airborne data; rather, we assume buildings are generally built vertically, and apply a histogram analysis on the angles between successive vertical scan points. If the average distribution peak is not centered at 90 degree, we set the roll angle estimate to the difference between histogram peak and 90 degree.

15.3.3. Processing Ground Based Data

At the end of the above steps, we obtain 6-DOF estimates for the global pose of the acquisition vehicle. This pose can be used to properly stack the vertical laser scans in order to construct a 3D point cloud of building facades, consisting of all the points in the vertical laser scans. As such, the points corresponding to the horizontal laser scans are not used in the actual model, and are only used for localization purposes. Since the resulting 3D point cloud is made of successive vertical scans, it has a special structure, which can be exploited in the triangulation step.

As described in [8], the path is segmented into easy to handle segments to be processed individually as follows: First, we remove foreground objects such as cars, trees, and pedestrians by first constructing a 2.5 D depth map from the resulting façade, and applying histogram analysis to separate foreground objects from background which consists of building facades [81]. This method works well in situations where there is sufficient separation between foreground and background. Next, we need to complete the 3D geometry, by filling the holes resulting either from removal of the foreground objects, or from objects such as glass which are transparent to the infrared laser scanner [81].

As photorealism cannot be achieved by using geometry alone, we need to enhance our model with texture data. To achieve this, we have equipped our data acquisition system with a digital color camera with a wide angle lens. The camera is synchronized with the two laser scanners, and is calibrated against the laser scanners’ coordinate system; hence the camera position can be computed for all images. After calibrating the camera and removing lens distortion in the images, each 3D vertex can be mapped to is corresponding pixel in an intensity image by a simple projective transformation. As the 3D mesh triangles are small compared to their distance to the camera, perspective distortions within a triangle can be neglected, and each mesh triangle can be mapped to a triangle in the picture by applying the projective transformation to its vertices.

Since foreground objects have been removed from 3D geometry, we need to remove the pixels corresponding to foreground objects from camera images in order to ensure they are not used to texture map background objects such as building facades. A simple way of segmenting out the foreground objects is to project the foreground mesh onto the camera images and mark out the projected triangles and vertices. While this process works adequately in most cases, it could potentially miss out some parts of the foreground objects [81]. This can be due to two reasons. First, the foreground scan points are not dense enough for segmenting the image with pixel accuracy especially at the boundary of foreground objects; second, the camera captures side views of foreground objects whereas the laser scanner captures a direct view. Hence some foreground geometry does not appear in the laser scans, and as such, cannot be marked as foreground.

To overcome this problem, we have developed a more sophisticated method for pixel accurate foreground segmentation based on the use of correspondence error and optical flow [81]. After segmentation, multiple images are combined into a single texture atlas. We have also developed a number of in-painting techniques to fill in the texture holes in the atlas resulting from foreground occlusion, including copy and paste and interpolation [81].

The result of a texture mapped façade model using the steps described in Section 15.3 is shown in Figure 15.6. Note that the upper parts of tall buildings are not texture-mapped, if they are outside the camera's field of view during data acquisition process. As will be seen in the next section, the upper parts of buildings are textured by airborne imagery. Alternatively, using multiple cameras could ensure that all parts of building facades are captured for texture mapping purposes.

Figure 15. 6: Ground-based facade models.

The texture for a path segment is typically several tens of megabytes, thus exceeding the rendering capabilities of today's graphics cards. Therefore, the facade models are optimized for rendering by generating multiple levels-of-detail (LOD), so that only a small portion of the entire model is rendered at the highest LOD at any given time. We subdivide the facade meshes along vertical planes and generate lower LODs for each sub-mesh, using the Qslim simplification algorithm [4] for geometry, and bi-cubic interpolation for texture reduction. All sub-meshes are combined in a scene graph, which controls the switching of the LODs depending on the viewer's position. This enables us to render the large amounts of geometry and texture with standard tools such as VRML browsers.

15.4 Model Merging

We now describe an approach to combine the ground-based facade models with the aerial surface mesh from the DSM. Both meshes are generated automatically, and given the complexity of a city environment, it is inevitable that some parts are partially captured, or completely erroneous, thus resulting in substantial discrepancies between the two meshes. Since our goal is a photorealistic virtual exploration of the city, creating models with visually pleasing appearances is more important than CAD properties such as water tightness. Common approaches for fusing meshes, such as sweeping and intersecting contained volume [11], or mesh zippering [13], require a substantial overlap between the two meshes. This is not the case in our application, since the two views are complementary. Additionally, the two meshes have entirely different resolutions: the resolution of the facade models, at about 10 to 15 cm, is substantially higher than that of the airborne surface mesh. Furthermore, to enable interactive rendering, it is required for the two models to fit together even when their parts are at different levels-of-detail.

Due to its higher resolution, it is reasonable to give preference to the ground-based facades wherever available, and use the airborne mesh only for roofs and terrain shape. Rather than replacing triangles in the airborne mesh for which ground-based geometry is available, we consider the redundancy before the mesh generation step in the DSM: for all vertices of the ground-based facade models, we mark the corresponding cells in the DSM. This is possible since ground-based models and DSM have been registered through the MCL localization techniques described in Section 15.3.3. We further identify and mark those areas in DSM, where our automated facade processing has classified as foreground such as trees and cars [8,81]. These marks control the subsequent airborne mesh generation from DSM; specifically, during the generation of the airborne mesh from DSM, (a) the z value for the foreground areas in DSM is replaced by the ground level estimate from the DTM, and (b) triangles at ground-based facade positions are not created. Note that the first step is necessary to enforce consistency and remove those foreground objects in the airborne mesh, which have already been deleted in the facade models. Figure 15.7(a) shows the DSM with facade areas marked in red and foreground marked in yellow, and Figure 15.7(b) shows the resulting airborne surface mesh with the corresponding façade triangles removed, and the foreground areas leveled to DTM altitude.

Figure 15.7: Removing facades from the airborne model; (a) marked areas in the DSM; (b) resulting mesh with corresponding facades and foreground objects removed. The arrows in (a) and (b) mark corresponding locations in DSM and mesh, respectively.

The facade models to be put in place do not match the airborne mesh perfectly, due to their different resolutions and capture viewpoints. Generally, the above procedure results in the removed geometry to be slightly larger than the actual ground–based facade to be placed in the corresponding location. To solve this discrepancy and to make mesh transitions less noticeable, we fill the gap with additional triangles to join the two meshes, and we refer to this step as “blending”. The outline of this procedure is shown in Figure 15.8. Our approach to creating such a blend mesh is to extrude the buildings along an axis perpendicular to the facades, as shown in Figure 15.8(b), and then shift the location of the “loose end” vertices to connect to the closest airborne mesh surface, as shown in Figure 15.8 (c). This is similar to the way plumb is used to close gaps between windows and roof tiles. These blend triangles are finally texture-mapped with the texture from the aerial photo, and as such, they attach at one end to the ground-based model, and at the other end to the airborne model, thus reducing visible seams at model transitions.

Figure 15.8: Creation of a blend mesh. A vertical cut through a building facade is shown. (a) Initial airborne and ground-based model registered; (b) facade of airborne model replaced and ground-based model extruded; (c) blending the two meshes by adjusting "loose ends" of extrusions to airborne mesh surface and mapping texture.

15.5. Results

We have applied the proposed algorithms on a data set for downtown Berkeley. Airborne laser scans have been acquired in conjunction with Airborne 1 Inc., at Los Angeles, CA; the entire data set consists of 60 million scan points. We have re-sampled these scan points to a 0.5 m × 0.5 m grid, and have applied the processing steps as described in Section 15.3 to obtain a DSM, an edge map, and a DTM for the entire area. We select feature points in five-mega pixel digital images taken from a helicopter and their correspondence in the DSM. This process takes about an hour for 12 images we use for the downtown Berkeley area. Then, the DSM is automatically triangulated, simplified, and finally texture-mapped. Figure 15.9(a) shows the surface mesh obtained from directly triangulating the DSM, 15.9(b) shows the triangulated DSM after the geometry processing steps outlined in Section 15.2.2, and 15.9(c) shows the texture-mapped model applying the texture processing steps outlined in Section 15.2.3. It is difficult to evaluate the accuracy of this airborne model, as no ground truth with sufficient accuracy is readily available, even at the city's planning department. We have admittedly sacrificed accuracy for the sake of visual appearance of the texture-mapped model, e.g. by removing small features on building tops. Thus, our approach combines elements of model-based and image-based rendering. While this is undesirable in some applications, we believe it is appropriate for interactive visualization applications.

Figure 15.9: Airborne model. (a) DSM directly triangulated, (b) triangulated after post-processing, (c) model texture-mapped.

The ground-based data has been acquired during two measurement drives in Berkeley: The first drive took 37 minutes long and was 10.2 kilometers long, starting from a location near the hills, going down Telegraph Avenue, and in loops around the central downtown blocks; the second drive was 41 minutes long and 14.1 kilometers, starting from Cory Hall at U.C. Berkeley and looping around the remaining downtown blocks. A total of 332,575 vertical and horizontal scans, consisting of 85 million scan points, along with 19,200 images, were captured during those two drives.

We first apply horizontal scan matching to provide an initial estimate of the driven path. We then correct that using MCL based technique of Section 15.3.3. In previous MCL experiments based on edge maps from aerial images with 30 cm resolution, we had found the localization uncertainty to be large at some locations, due to false edges and perspective shifts; to overcome this, we have had to use large number of particles, e.g. 120,000 with MCL in order to approximate the spread-out probability distribution appropriately, and track the vehicle reliably. Even though the edge map derived from airborne laser scans described in Section 15.3.2 has a lower resolution than an edge map resulting from aerial imagery, we can track the vehicle using DSM with as few as 5000 particles. This is because the edge map derived from DSM is more accurate and reliable than that of aerial imagery.

Having computed an initial path estimate using horizontal scan matching, we apply global MCL based correction based on DSM edge maps to the yaw angles in path 1 as shown in Figure 15.10 (a). We then re-compute the path and apply the correction to the x and y coordinates, as shown in Figure 15.10 (b). As expected, the global correction substantially modifies the initial pose estimates, thus reducing errors in subsequent processing steps. Figure 15.10 (c) plots the assigned z coordinate, clearly showing the slope from our starting position at higher altitude near the Berkeley Hills down towards the San Francisco Bay, as well as the ups and downs on this slope while looping around the downtown blocks.

Figure 15.10: MCL Global correction for path 1 using DSM based edge maps; (a) yaw angle difference between initial path and global estimates before and after correction; (b) differences of x and y coordinates before and after correction; (c) assigned z coordinates. In plots (a) and (b), the differences after corrections are the curves close to the horizontal axis.

Figure 15.11 (a) shows uncorrected paths 1 and 2 superimposed on the airborne DSM using horizontal scan matching. Figure 15.11 (b) shows the paths after MCL global correction based on DSM. As seen, the corrected path coincides nicely with the Manhattan structure of the roads in the DSM, whereas the uncorrected one is clearly erroneous as it is seen to cross over parts of DSM corresponding to buildings. Figure 15.12 shows the ground based horizontal scan points for the corrected paths superimposed on top of DSM. As expected, horizontal scan points appear to closely match the DSM edges for the corrected paths.

Figure 15.11: Driven paths superimposed on top of the DSM (a) before correction; (b) after correction. The circles denote the starting position for paths 1 and 2, respectively.

Figure 15.12: Horizontal scan points for corrected paths.

After MCL correction, all scans and images are geo-referenced. This is used to properly stack vertical laser scans in order to generate a 3D point cloud which is then triangulated, processed and texture mapped using the steps described in Section 15.3.4. Figure 15.13 shows the resulting facades for the 12 street blocks of downtown Berkeley. Note that the acquisition time for the 12 downtown Berkeley blocks has been only 25 minutes; this is the time it took to drive of both paths that it took to drive the total of 8 kilometers around these 12 blocks under city traffic conditions.

Figure 15.13: Façade model for downtown Berkeley Area.

Due to the usage of the DSM as the global reference for MCL, the DSM and facade models are registered with each other, and we can apply the model merging steps as described in Section IV. Figure 15.14 (a) shows the resulting combined model for the downtown Berkeley blocks, as viewed in a walk-thru or drive-thru, and 15.14(b) shows a view from the rooftop of a downtown building. Due to the limited field of view of the ground-based camera, the upper parts of the building facades are texture-mapped with aerial imagery. The noticeable difference in resolution between the upper and lower parts of the texture on the building in Figure 15.14(b) emphasizes the necessity of ground-based facade models for walk-thru applications. Future systems should use multiple ground based cameras in order to ensure capturing all parts of all buildings. This would of course be at the expense of a significant increase in data collection volume. Figure 15.15 shows the same model as Figure 5.14(b) in a view from the top, as it appears in a fly-thru. The model can be downloaded for interactive visualization from the website in [15].

Figure 15.14: Walk-thru view of the model; (a) as seen from the ground level; (b) as seen from the rooftop of a building.

Figure 15.15: Bird's eye view of the model.

Our proposed approach to city modeling is not only automated, but also fast from a computational viewpoint: As shown in Table 15.1, the total time for the automated processing and model generation for the twelve downtown blocks is around five hours on a 2 GHz Pentium-4 PC. Since the complexity of all developed algorithms is linear in area and path length, our method is scalable to large environments.

Table 15.1: Processing times for the downtown Berkeley blocks.

15.6. Applications of 3D Modeling to Surveillance

A priori knowledge of the 3D scene in which a target is located, or “on the fly” 3D reconstruction of the scene, can be tremendously helpful in identifying and tracking it in surveillance applications. Consider Figure 15.16 where a helicopter is looking for and tracking a person of interest on the ground in downtown Berkeley. If the 3D model of the target of interest is available at the helicopter, the location, orientation, and velocity of the target can be easily determined by matching the output of the image sensors, or other 3D sensors focused on the target, with the 3D model of the scene surrounding it. Similar arguments can be made if ground based surveillance cameras were randomly scattered on streets of a city, where each camera station has access to complete 3D map of its own surrounding environment. To begin with, the a priori textured 3D model of the surrounding environment, together with simple background/foreground separation techniques can be readily used to separate moving objects within the scene from the stationary background already captured in the city modeling process. Once target identification is carried out, a priori knowledge of the textured 3D structures of the city areas in which the target is moving in, can be used to facilitate target localization and tracking process, regardless of whether it is done by a single sensor which “moves” with the target or by multiple cameras that “hand off” the tracking process to each other as the target moves from one field of view to the other. In addition, the knowledge of the geometry of the scene can be used to optimally and adaptively adjust the pan, zoom, tilt and other parameters of the cameras during the co-location and tracking processes. Finally, a priori knowledge of the 3D textured models of a scene can be readily used to register multiple tracking cameras with respect to each other. Textured 3D models of the scene can also be used in conjunction with spatio-temporal super resolution techniques from signal processing to enhance the ability to locate targets with sub-pixel accuracy and more frequent temporal updates, along similar lines shown in [77].

Figure 15.16: Illustration of ways in which a priori models of 3D cities can be used for surveillance applications.

15.7. Summary and Conclusions

We have proposed a fast, automated system for acquiring 3D models of urban environments for virtual walk-throughs, drive-thoughs and fly-throughs. The basic approach has been to use ground based and aerial laser scanners and cameras in order to construct ground and aerial models. The DSM resulting from airborne modeling is used in conjunction with MCL technique, not only to localize the ground based acquisition vehicle accurately, but also to register the ground and aerial models. Future work involves developing fast, automated approaches to texture mapping airborne models using oblique aerial imagery as currently this is the only compute intensive step in our end to end processing. Even though this step can be sped up significantly using semi-automatic approach, it is highly desirable to keep the human out of the loop during the model construction process. We also plan to explore the use of 3D models in surveillance and tracking applications.

15.8. Acknowledgements

The author would like to acknowledge Rusty Sammon, Siddarth Jain, John Secord, John Flynn, Steve Jian, John Lo, Dr. Hassan Foroosh, Ali Lakhia, and all other Berkeley students and staff who have participated in this project over the past six years. This work was sponsored by Army Research Office contract DAAD 19-00-1-0352, under a MURI program.

15.9. References

[1] H. Araujo, R. L. Carceroni, and C. M. Brown, “A Fully Projective Formulation to Improve the Accuracy of Lowe's Pose-Estimation Algorithm”, Computer Vision and Image Understanding, Vol. 70, No. 2, pp. 227-238, May 1998

[2] C. Brenner, N. Haala, and D. Fritsch: “Towards fully automated 3D city model generation”, Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images III, 2001

[3] A. Dick, P. Torr, S. Ruffle, and R. Cipolla, “Combining Single View Recognition and Multiple View Stereo for Architectural Scenes”, Int. Conference on Computer Vision, Vancouver, Canada, 2001, p. 268-74

[4] M. Garland and P. Heckbert, “Surface Simplification Using Quadric Error Metrics”, SIGGRAPH ‘97, Los Angeles, 1997, p. 209-216

[5] M. A. Fischler and R. C. Bolles: ”Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography”, Communication Association and Computing Machine, 24(6), pp.381-395, 1981

[6] C. Früh and A. Zakhor, ”Fast 3D model generation in urban environments”, IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems, Baden-Baden, Germany, 2001, p. 165-170

[7] C. Früh and A. Zakhor, ”3D model generation of cities using aerial photographs and ground level laser scans”, Computer Vision and Pattern Recognition, Hawaii, USA, 2001, p. II-31-8, vol.2. 2001.

[8] C. Früh and A. Zakhor, “Data Processing Algorithms for Generating Textured 3D Building Façade Meshes From Laser Scans and Camera Images”, 3D Processing, Visualization and Transmission 2002, Padua, Italy, 2002, p. 834 – 847

[9] Z. Kim, A. Huertas, and R. Nevatia, “Automatic description of Buildings with complex rooftops from multiple images”, Computer Vision and Pattern Recognition, Kauai, 2001, p. 272-279

[10] V. Sequeira, J.G.M. Goncalves, ”3D reality modeling: photo-realistic 3D models of real world scenes,” Proc. First International Symposium on 3D Data Processing Visualization and Transmission 2002, pp 776 –783

[11] I. Stamos and P. K. Allen, “Geometry and Texture Recovery of Scenes of Large Scale”, Computer Vision and Image Understanding (CVIU), V. 88, N. 2, Nov. 2002, pp. 94-118.

[12] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, “Robust Monte Carlo Localization for Mobile Robots”, Artificial Intelligence, 128 (1-2), 2001

[13] G. Turk and M. Levoy, “Zippered Polygon Meshes from Range Images”, SIGGRAPH '94, Orlando, Florida, 1994, pp. 311-318.

[14] H. Zhao, R. Shibasaki, “Reconstructing a textured CAD model of an urban environment using vehicle-borne laser range scanners and line cameras,” Machine Vision and Applications 14 (2003) 1, pp. 35-41

[15] http://www-video.eecs.berkeley.edu/~frueh/3d/

[16] M.E. Antone and S. Teller: “Automatic recovery of relative camera rotations for urban scenes. “ Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, 2000, p.282-9

[17] P. Besl and N. McKay: “A Method for Registration of 3-D Shapes”, IEEE Transactions on PAMI, Vol. 14, No. 2, 1992.

[18] W. Burgard, D. Fox, D. Hennig, and T. Schmidt: "Estimating the Absolute Position of a Mobile Robot Using Position Probability Grids", AAAI, 1996.

[19] R. Chan, W. Jepson, and S. Friedman: “Urban Simulation: An Innovative Tool for Interactive Planning and Consensus Building”, Proceedings of the 1998 American Planning Association National Conference, Boston, MA pages 43-50, 1998

[20] I. J. Cox: “Blanche - An experiment in guidance and navigation of an autonomous robot vehicle”, IEEE Transactions on Robotics and Automation, vol 7, pp 193-204, 1991

[21] P. E. Debevec, C. J. Taylor, and J. Malik: “Modeling and Rendering Architecture from Photographs”, Proc. of ACM SIGGRAPH 1996

[22] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun: "Structure from Motion without Correspondence", IEEE Conference on Computer Vision and Pattern Recognition, 2000

[23] D. Fox, W. Burgard, S. Thrun: “Markov Localization for Mobile Robots in Dynamic Environments”, Journal of Artificial Intelligence Research 11, pp. 391-427, 1999

[24] D. Fox, S. Thrun, F. Dellaert, and W. Burgard: “Particle filters for mobile robot localization”, In A. Doucet, N. de Freitas, and N. Gordon, edts, Sequential Monte Carlo Methods in Practice. Springer Verlag, New York, 2000

[25] C. Früh: “Automated 3D Model Generation for Urban Environments”, PhD Thesis, University of Karlruhe, Germany, 2002

[26] D. Frere, J. Vandekerckhove, T. Moons, and L. Van Gool: “Automatic modelling and 3D reconstruction of urban buildings from aerial imagery”, IEEE International Geoscience and Remote Sensing Symposium Proceedings, Seattle, 1998, p.2593-6

[27] J.-S. Gutmann and K. Konolige: “Incremental Mapping of Large Cyclic Environments”, International Symposium on Computational Intelligence in Robotics and Automation (CIRA'99), Monterey, 1999

[28] J.-S. Gutmann and C. Schlegel: “Amos: Comparison of scan matching approaches for

self-localization in indoor environments", Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots, 1996

[29] D. Hähnel, W. Burgard, and S. Thrun: "Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot", 4th European Workshop on Advanced Mobile Robots (EUROBOT'01), 2001.

[30] A. Huertas, R. Nevatia, and D. Landgrebe: "Use of hyperspectral data with intensity images for automatic building modeling", Proc. of the Second International Conference on Information Fusion, Sunnyvale, 1999, p.680-7 vol.2. 2

[31] P. Jensfelt and S. Kristensen: "Active global localization for a mobile robot using multiple hypothesis tracking", Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robot Navigation, pages 13–22, Stockholm, Sweden, 1999. IJCAI

[32] H. Kawasaki, T. Yatabe, K. Ikeuchi, and M. Sakauchi: "Automatic modeling of a 3D city map from real-world video", Proceedings ACM Multimedia, Orlando, USA, 1999, p. 11-18

[33] S. Koenig and R. Simmons: "A robot navigation architecture based on partially observable Markov decision process models", In Kortenkamp et al., 1998

[34] K. Konolige and K. Chou: "Markov Localization using Correlation", Proc. International Joint Conference on Artificial Intelligence (IJCAI'99), Stockholm, 1999.

[35] F. Lu and E. Milios: "Robot pose estimation in unknown environments by matching 2D range scans", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1994

[36] F. Lu and E. Milios: "Robot pose estimation in unknown environments by matching 2D range scans", Journal of Intelligent and Robotic Systems, 18, 1997

[37] F. Lu and E. Milios: "Globally Consistent Range Scan Alignment for Environment Mapping", Autonomous Robots 4, pp. 333 - 349, 1997

[38] H.-G. Maas, "The suitability of airborne laser scanner data for automatic 3D object reconstruction", Third Int'l Workshop on Automatic Extraction of Man-Made Objects, Ascona, Switzerland, 2001

[39] S.I. Roumeliotis and G.A. Bekey: "Bayesian estimation and Kalman filtering: A unified framework for mobile robot localization", Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pages 2985–2992, San Francisco, CA, 2000

[40] S. J. Russell and P. Norvig: “Articial Intelligence: A Modern Approach”, Chapter 17, Series in Articial Intelligence, Prentice Hall, 1995

[41] S. Seitz and C. Dyer, "Photorealistic scene reconstruction by voxel coloring", Proc. CVPR 1997, pp. 1067-1073, 1997.

[42] R. Simmons and S. Koenig: ”Probabilistic robot navigation in partially observable environments”, Proc. of International Joint Conference on Artificial Intelligence, Montreal, 1995. p.1080-7 vol.2

[43] I. Stamos and P.E. Allen: “3-D model construction using range and image data.” Proceedings IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, 2000, p.531-6

[44] R. Szeliski and S. Kang, "Direct methods for visual scene reconstruction", IEEE Workshop on Representation of Visual Scenes, pages 26-33, June 1995

[45] S. Thrun: “Probabilistic algorithms in robotics”, AI Magazine, vol.21, American Assoc. Artificial Intelligence, Winter 2000, p. 93-109

[46] S. Thrun, W. Burgard, and D. Fox: “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping”, Proc. of ICRA 2000, San Francisco, 2000, p.321-8, vol. 1. 4

[47] S. Thrun: "A probabilistic online mapping algorithm for teams of mobile robots", International Journal of Robotics Research, 2001, no.5, vol. 20, p. 335-363.

[48] B. Triggs, P.F. McLauchlan, R.I. Hartley, A.W. Fitzgibbon: “Bundle adjustment - a modern synthesis”, Proc. International Workshop on Vision Algorithms, Corfu, 2000, p. 298-372

[49] G. Weiss, C. Wetzler, and E. von Puttkamer: “Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans”, Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1994

[50] H. Zhao, R. Shibasaki: "A System for Reconstructing Urban 3D Objects using Ground- Based Range and CCD Images", Proc. of International Workshop on Urban Multi-Media/3D Mapping, Tokyo, 1999

[51] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro and J. Verdera, “Filling in by joint interpolation of vector fields and gray levels”, IEEE Transactions on Image Processing August 2001, p. 1200-1211.

[52] C. Ballester, V. Caselles, J. Verdera, M. Bertalmio and G. Sapiro, “A variational model for filling-in gray level and color images”, Proc. 8th IEEE Int’l Conference on Computer Vision, p. 10-16. vol. 1

[53] M. Bertalmio, G. Sapiro, C. Ballester and V. Caselles, “Image Inpainting”, Proc. SIGGRAPH 2000, p. 417-424

[54] T. Chan and J. Shen, “Mathematical models for local nontexture inpaintings”, SIAM Journal on Applied Mathematics 2001, p. 1019-1043, vol 62 number 3.

[55] N.L. Chang and A. Zakhor, “A Multivalued Representation for View Synthesis”, Proc. Int’l Conference on Image Processing, Kobe, Japan, 1999, vol. 2, pp. 505-509

[56] B. Curless and M. Levoy, “A volumetric method for building complex models from range images”, SIGGRAPH, New Orleans, 1996, p. 303-312

[57] P.E. Debevec, C. J. Taylor and J. Malik, “Modeling and Rendering Architecture from Photographs”, Proceedings of ACM, SIGGRAPH 1996

[58] A. Dick, P. Torr, S. Ruffle, and R. Cipolla, “Combining Single View Recognition and Multiple View Stereo for Architectural Scenes”, International Conference on Computer Vision, Vancouver, Canada, 2001, p. 268-74

[59] A. Efros and W. Freeman, “Image quilting for texture synthesis and transfer”, Proc. SIGGRAPH 2001, p. 341-346.

[60] J.D. Foley, A. van Dam, S.K. Feiner and J.F. Hughes, “Computer Graphics: Principles and Practice” second edition in C, Section 19.5, Addison-Wesley, Reading MA 1995.

[61] D.A. Forsyth and J. Ponce, “Computer Vision – a modern approach”, chapter 7. Prentice- Hall, 2002.

[62] D. Frere, J. Vandekerckhove, T. Moons, and L. Van Gool, “Automatic modelling and 3D reconstruction of urban buildings from aerial imagery”, IEEE International Geoscience and Remote Sensing Symposium Proceedings, Seattle, 1998, p.2593-6

[63] M. Garland and P. Heckbert, “Surface Simplification Using Quadric Error Metrics”, SIGGRAPH ‘97, Los Angeles, 1997, p. 209-216

[64] N. Haala and C. Brenner, “Generation of 3D city models from airborne laser scanning data”, Proc. EARSEL workshop on LIDAR remote sensing on land and sea, Tallin, Esonia, 1997, p.105-112

[65] Z. Kim, A. Huertas, and R. Nevatia, “Automatic description of Buildings with complex rooftops from multiple images”, Computer Vision and Pattern Recognition, Kauai, 2001, p. 272-279

[66] H.G. Maas, “The suitability of airborne laser scanner data for automatic 3D object reconstruction”, 3. Int’l Workshop on Automatic Extraction of Man-Made Objects, Ascona, Switzerland, 2001

[67] S. Masnou and J. Morel, “Level-lines based disocclusion”, 5th IEEE Int’l Conference on Image Processing, 1998, p. 259-263.

[68] M. Nitzberg., D. Mumford and T. Shiota, “Filtering, Segmentation and Depth”, Springer Verlag, Berlin 1993.

[69] S. Thrun, W. Burgard, and D. Fox, “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping”, Proc. of International Conference on Robotics and Automation, San Francisco, 2000, p..321-8, vol. 1. 4

[70] C. Vestri and F. Devernay, “Using Robust Methods for Automatic Extraction of Buildings”, Computer Vision and Pattern Recognition, Hawaii, USA, 2001, p. I-133-8, vol.1. 2

[71] X. Wang, S. Totaro, F. Taillandier, A. Hanson, and S. Teller, “Recovering Facade Texture and Microstructure from Real-World Images”, Proc. 2nd International Workshop on Texture Analysis and Synthesis in conjunction with European Conference on Computer Vision, June 2002, pp. 145—149.

[72] J. Davis, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime Stereo: A Unifying Framework for Depth from Triangulation. 2003 Conference on Computer Vision and Pattern Recognition (CVPR 2003). pp. 359-366, June 2003.

[73] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, H. Fuchs. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. Computer Graphics Proceedings, SiGGRAPH 1998, Orlando, FL.

[74] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-Time 3D Model Acquisition. ACM Transactions on Graphics, 21(3):438-446, 2002.

[75] J. Salvi, J. Pagès, and J. Batlle. Pattern codification strategies in structured light systems. Pattern Recognition, Article 1971, (2003).

[76] J. Carranza, C. Theobalt. M. Magnor, H.P. Seidel. Free-Viewpoint Video of Human Actors. Proc. of ACM, SIGGRAPH 2003, p. 569-577, San Diego, CA.

[77] N. S. Shah and A. Zakhor, ``Resolution Enchancement of Color Video Sequences,**"** *IEEE Transactions on Image Processing*, June 1999, vol. 8, no. 6, pp. 879-885.

[78] J. Hu, S. You, and U. Neumann, “Approaches to Large Scale Urban Modleing”, IEEE Computer Graphics and Applications, Volume 23, No. 6, pp. 62 – 69, November 2003.

[79] L. Wang, S. You, and U. Neumann, “Large Scale Urban Modeling by Combining Ground Level

Panoramic and Aerial Imagery”, 3DPTV, June 2006, Chapel Hill, USA.

[80] C. Frueh, R. Sammon, and A. Zakhor, “Automated Texture Mapping of 3D City Models with Oblique Aerial Imagery”, in Second International Symposium on 3D Data Processing, Visualization, and Transmission, Thessaloniki, Greece, September 2004, pp. 396-403.

[81] C. Frueh, S. Jain, and A. Zakhor, “Data Processing for Generating Textured 3D Building Façade Meshes from Laser Scans and Camera Images”, International Journal of Computer Vision, Vol. 61, No. 2, Feb. 2005, pp. 159-184.

[82] C. Frueh and A. Zakhor, “An automated Method for Large Scale Ground Based City Model Acquisition” International Journal of Computer Vision, Vol. 60, No. 1, October 2004, pp. 5 – 24.

[83] C. Frueh and A. Zakhor "Constructing 3D City Models by Merging Ground-Based and Airborne Views" in *Computer Graphics and Applications*, November/December 2003, pp. 52 - 61.

[84] C. Frueh and A. Zakhor, "Reconstructing 3D City Models by Merging Ground-Based and Airborne Views" in Proceedings of the 8th International Workshop VLBV 2003, Madrid, Spain, September 2003, pp. 306-313.

[85] C. Frueh and A. Zakhor, "Automated Reconstruction of Building Façades for Virtual Walk-thrus" in SIGGRAPH Sketches and Applications, San Diego, July 2003.

[86] C. Frueh and A. Zakhor, "Constructing 3D City Models by Merging Ground-Based and Airborne Views" in Conference on Computer Vision and Pattern Recognition 2003, Madison, Wisconsin, June 2003.

[87] J. Secord and A. Zakhor, "Tree Detection In Aerial LiDAR and Image Data", International Conference on Image Processing, Atlanta, Georgia, September 2006.

[88] J. Secord and A. Zakhor, "Tree Detection In Aerial LiDAR Data", Southwest Symposium on Image Analysis and Interpretation Denver, Colorado, March 2006.

[89] R. Nevatia and A. Huertas, “Knowledge Based Building Detection and Description: 1997-1998”, R. Nevatia and A. Huertas, Proceedings of the Darpa Image Understanding Workshop, Monterey, CA 1998.

[90] AJ Heller, MA Fischler, RC Bolles, CI Connolly, “An Integrated feasibility demonstration for automatic population of geospatial databases”, Proceedings of the Darpa Image Understanding Workshop, Monterey, CA, 1998.

[91] “RADIUS: Image Understanding for Imagery Intelligence”, Edited by O. Firshchein and T. Strat, Morgan Kaufman, San Mateo, CA, 1997.

[92] “Automatic Extraction of Objects from Aerial and Space Images” Editted by A. Gruen, E.P. Baltsavias, and O. Henricsson; Birkhauser, 1997.

[93] A. Akbarzadeh, J.-M. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, H. Towles, D. Nister and M. Pollefeys, *Towards Urban 3D Reconstruction From Video*, Proc. 3DPVT'06 (Int. Symp. on 3D Data, Processing, Visualization and Transmission), 2006.

[94] M. Pollefeys, “*3D from Image Sequences: Calibration, Motion and Shape Recovery”*, Mathematical Models of Computer Vision: The Handbook, N. Paragios, Y. Chen, O. Faugeras, Springer, 2005.

[95] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, “*Visual modeling with a hand-held camera”*, International Journal of Computer Vision Vol. 59, No. 3, pp. 207-232, 2004.

[96] D. Nister, “Automatic passive recovery of 3D from images and video”, Proceedings of the second international symposium on 3D Data Processing, Visualization and Transmission, 3DPTV, September 2004, pp. 438-445.

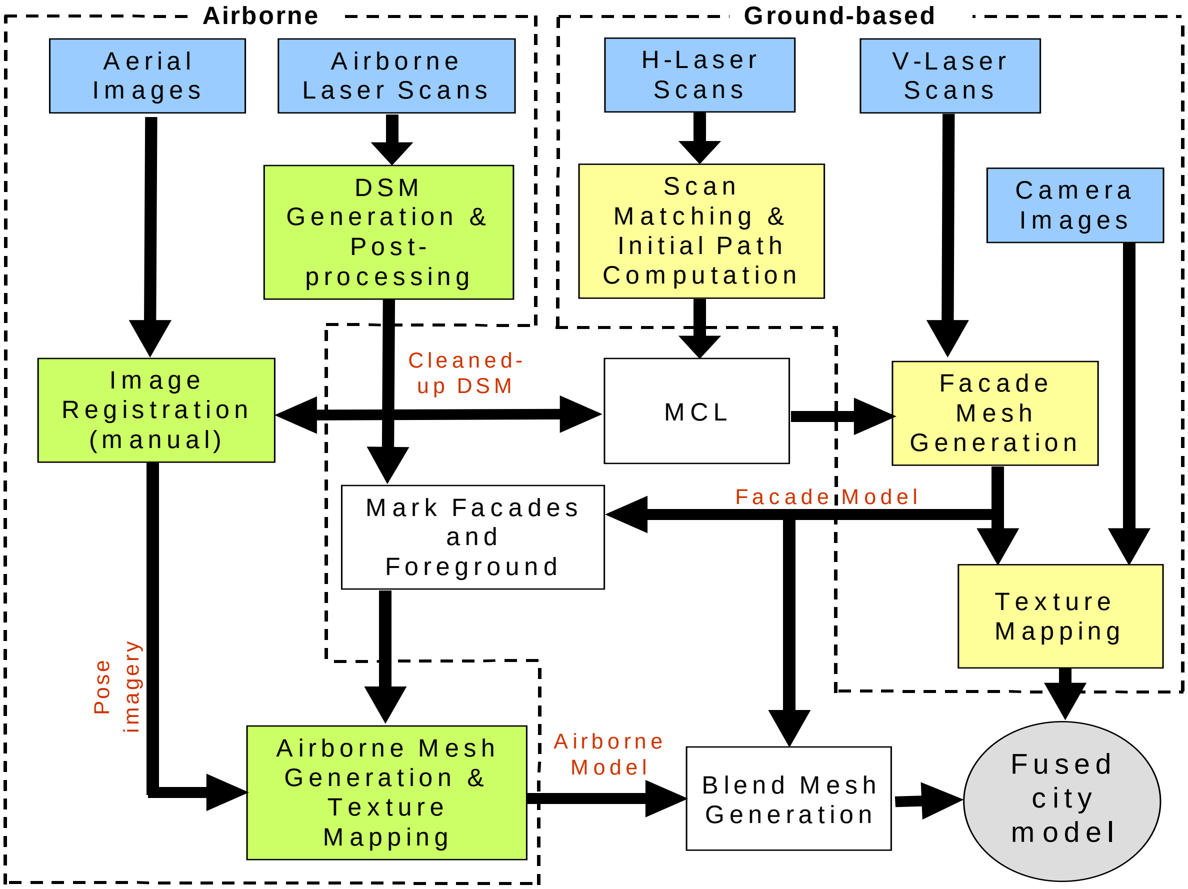
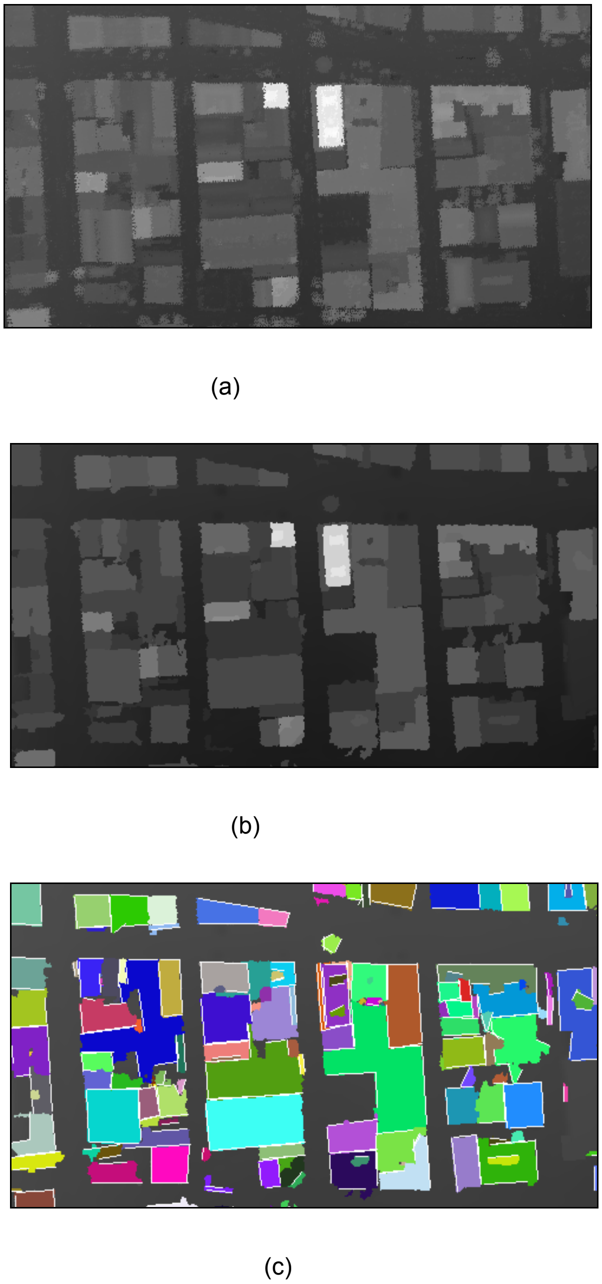


Figure 15.1: Data flow diagram of our modeling approach. Airborne modeling steps are highlighted in green, ground-based modeling steps in yellow, and model fusion steps in white.



(a)

(b)

(c)

Figure 15.2: Processing steps for DSM; (a) DSM obtained from scan point re-sampling; (b) DSM after flattening roofs; (c) segments with RANSAC lines in white.

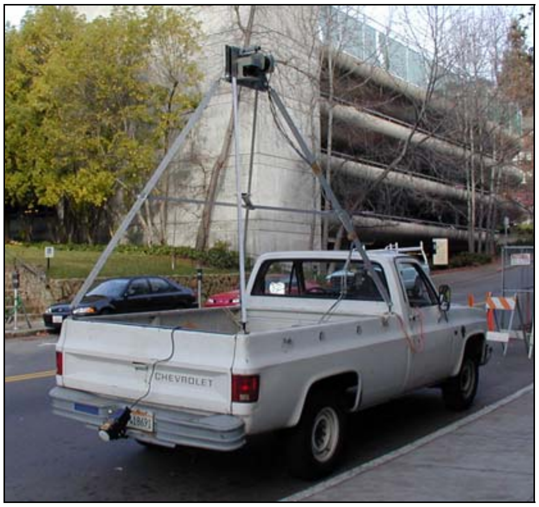


Figure 15.3: Acquisition vehicle.

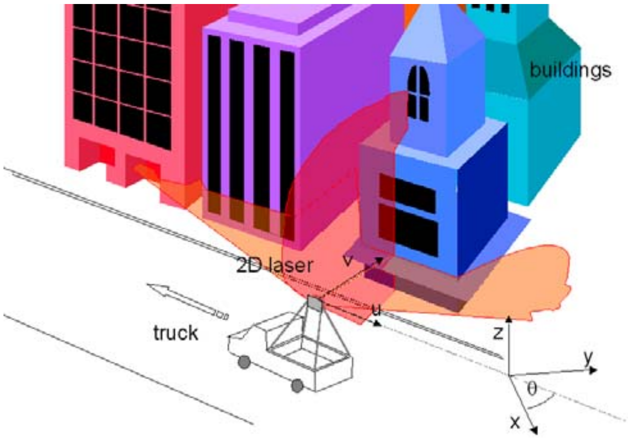
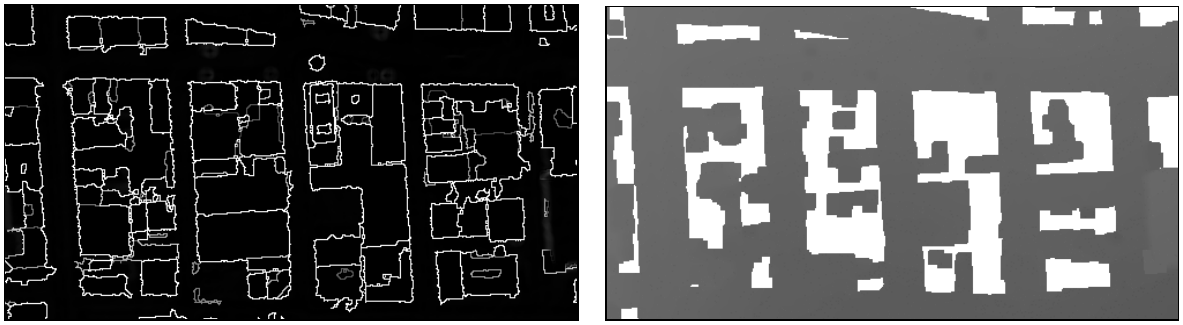


Figure 15.4: Ground-based acquisition setup.



(a) (b)

Figure 15.5: Map generation for MCL; (a) edge map; (b) DTM. For the white pixels, there is no ground level estimate available.

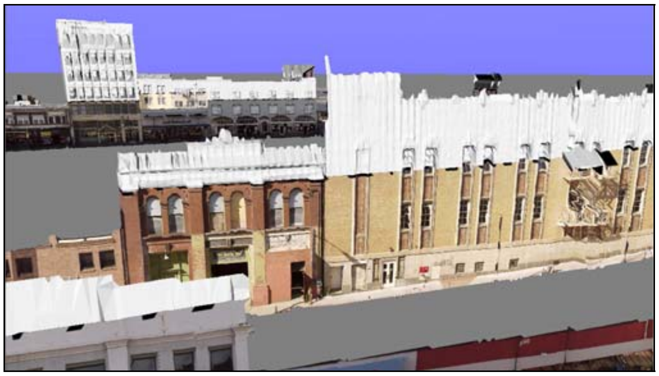
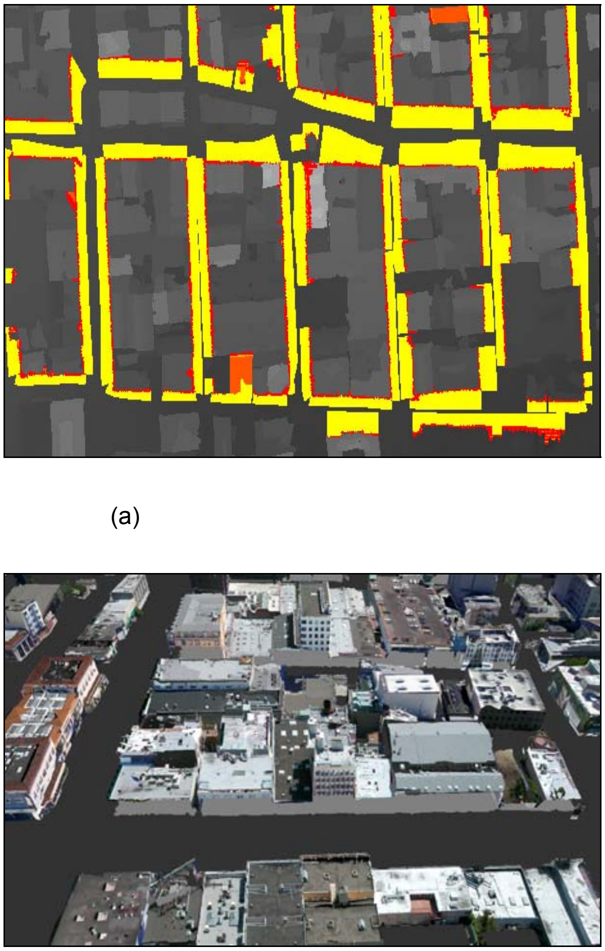


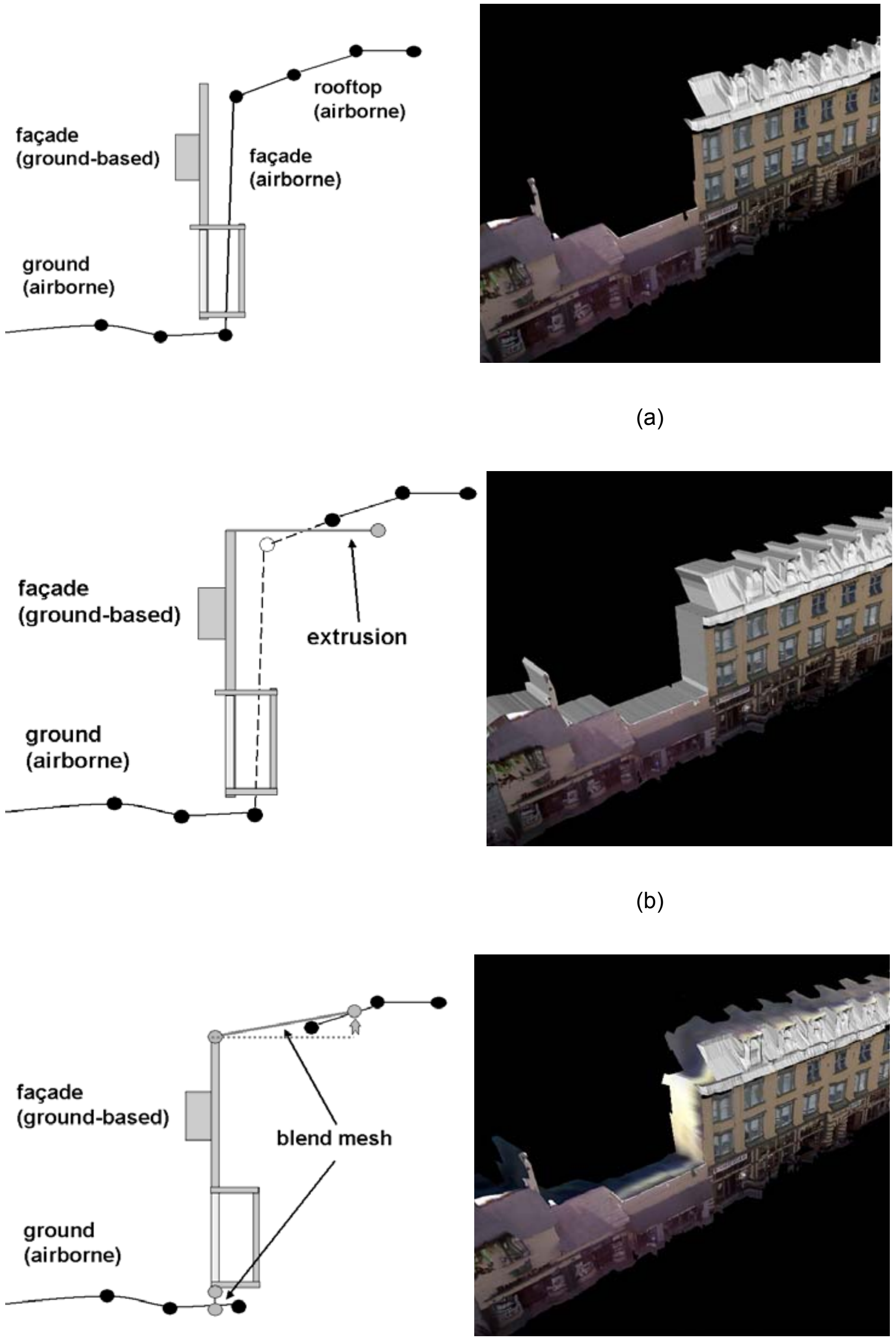
Figure 15. 6: Ground-based facade models.



(a)

(b)

Figure 15.7: Removing facades from the airborne model; (a) marked areas in the DSM; (b) resulting mesh with corresponding facades and foreground objects removed. The arrows in (a) and in (b) mark corresponding locations in DSM and mesh, respectively.

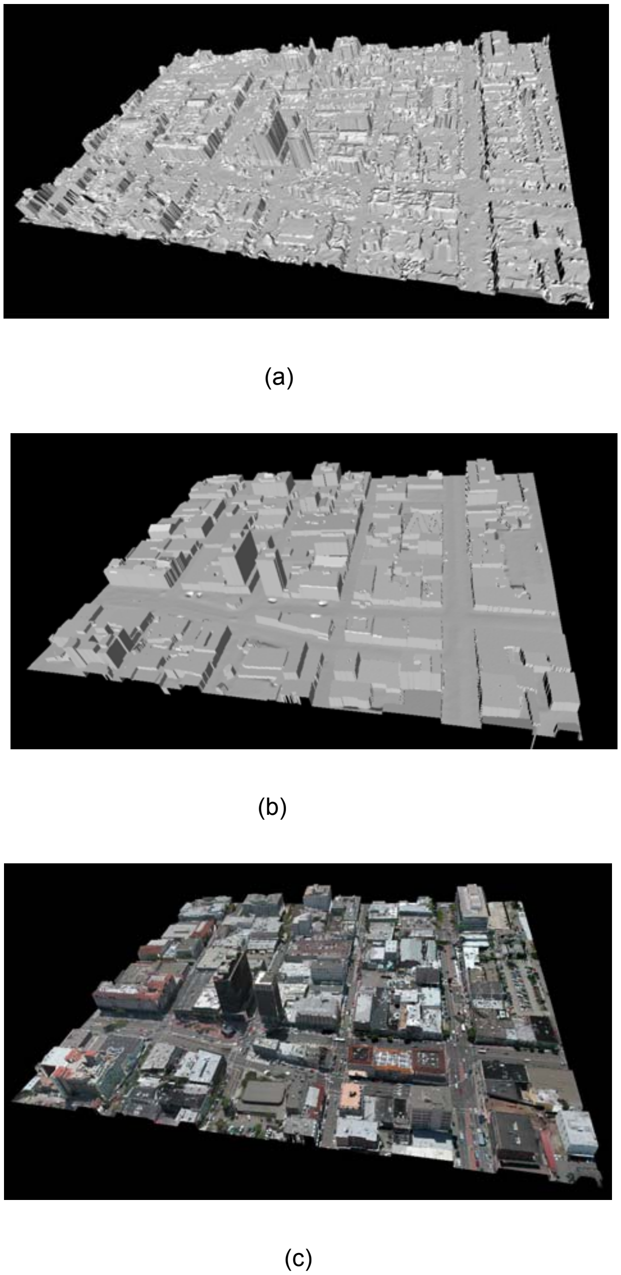


(a)

(b)

(c)

Figure 15.8: Creation of a blend mesh. A vertical cut through a building facade is shown. (a) Initial airborne and ground-based model registered; (b) facade of airborne model replaced and ground-based model extruded; (c) blending the two meshes by adjusting "loose ends" of extrusions to airborne mesh surface and mapping texture.

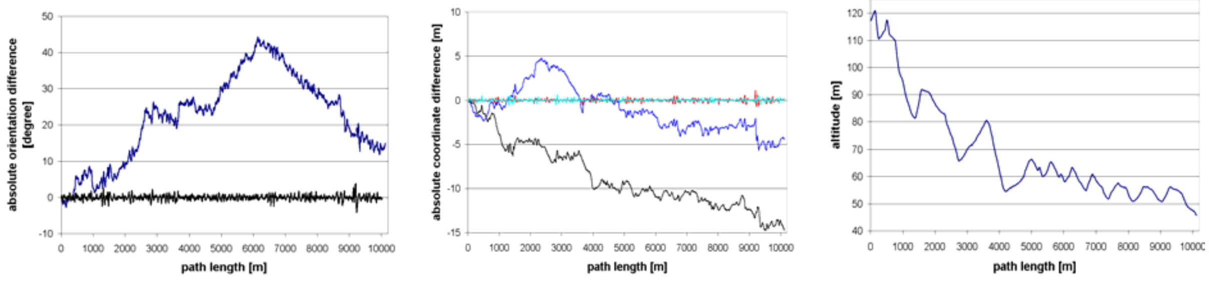


(a)

(b)

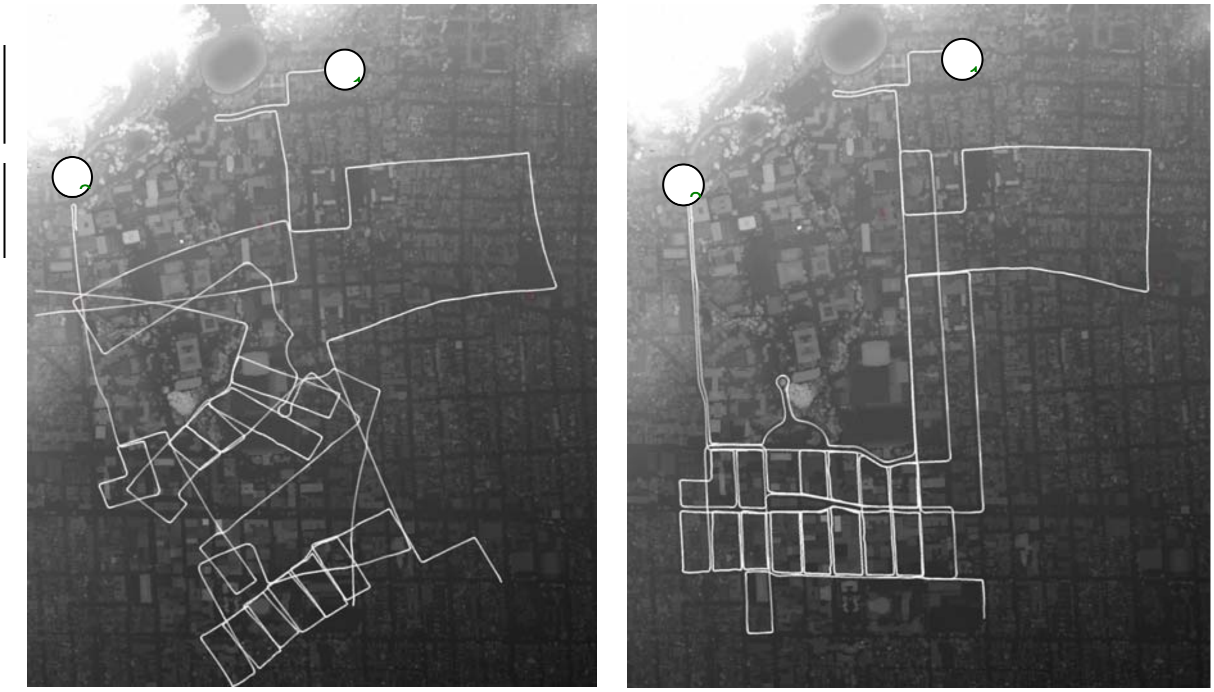
(c)

Figure 15.9: Airborne model. (a) DSM directly triangulated, (b) triangulated after postprocessing, (c) model texture-mapped.



(a) (b) (c)

Figure 15.10: Global correction for path 1; (a) yaw angle difference between initial path and global estimates before and after correction; (b) differences of x and y coordinates before and after correction; (c) assigned z coordinates. In plots (a) and (b), the differences after corrections are the curves close to the horizontal axis.



(a) (b)

Figure 15.11: Driven paths superimposed on top of the DSM (a) before correction, and (b) after correction. The circles denote the starting position for paths 1 and 2, respectively.

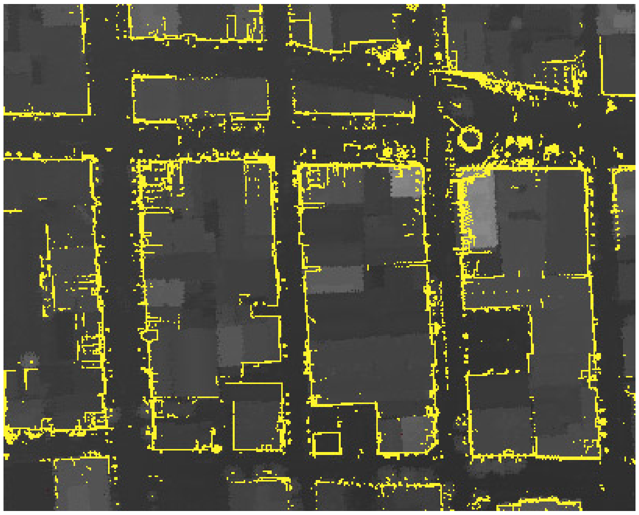


Figure 15.12: Horizontal scan points for corrected paths.

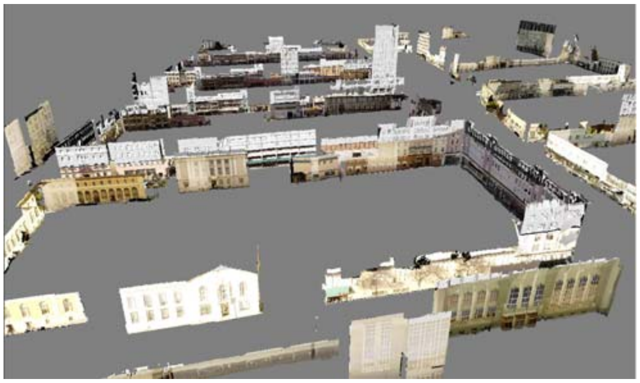
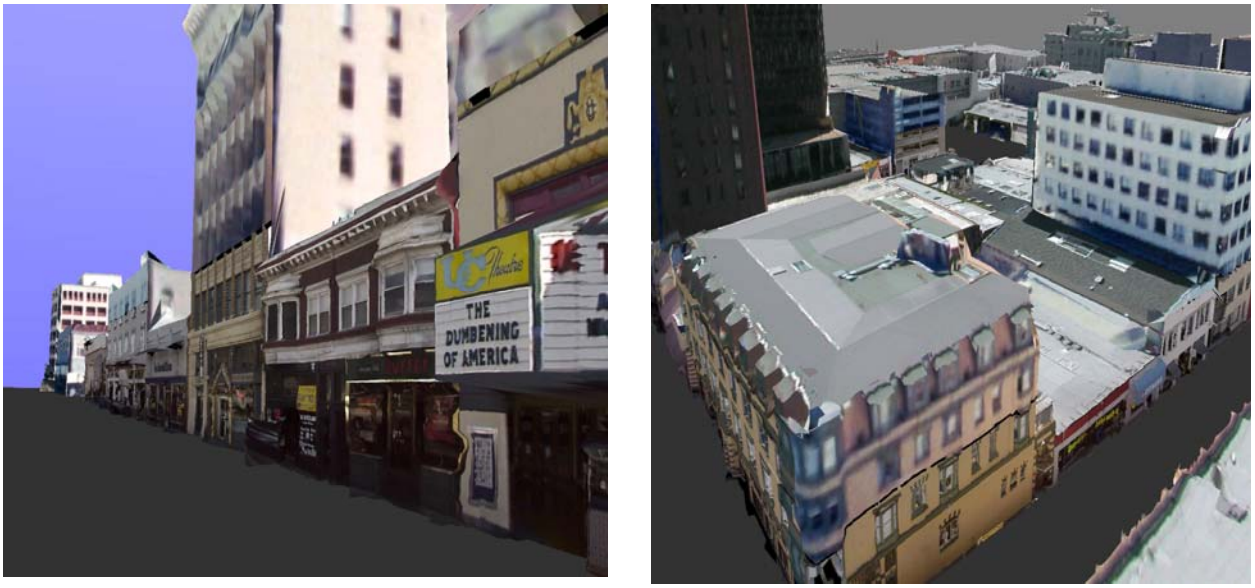


Figure 15.13: Façade model for downtown Berkeley Area



(a)

(b)

Figure 15.14: Walk-thru view of the model; (a) seen from the ground level; (b) seen from the rooftop of a building.

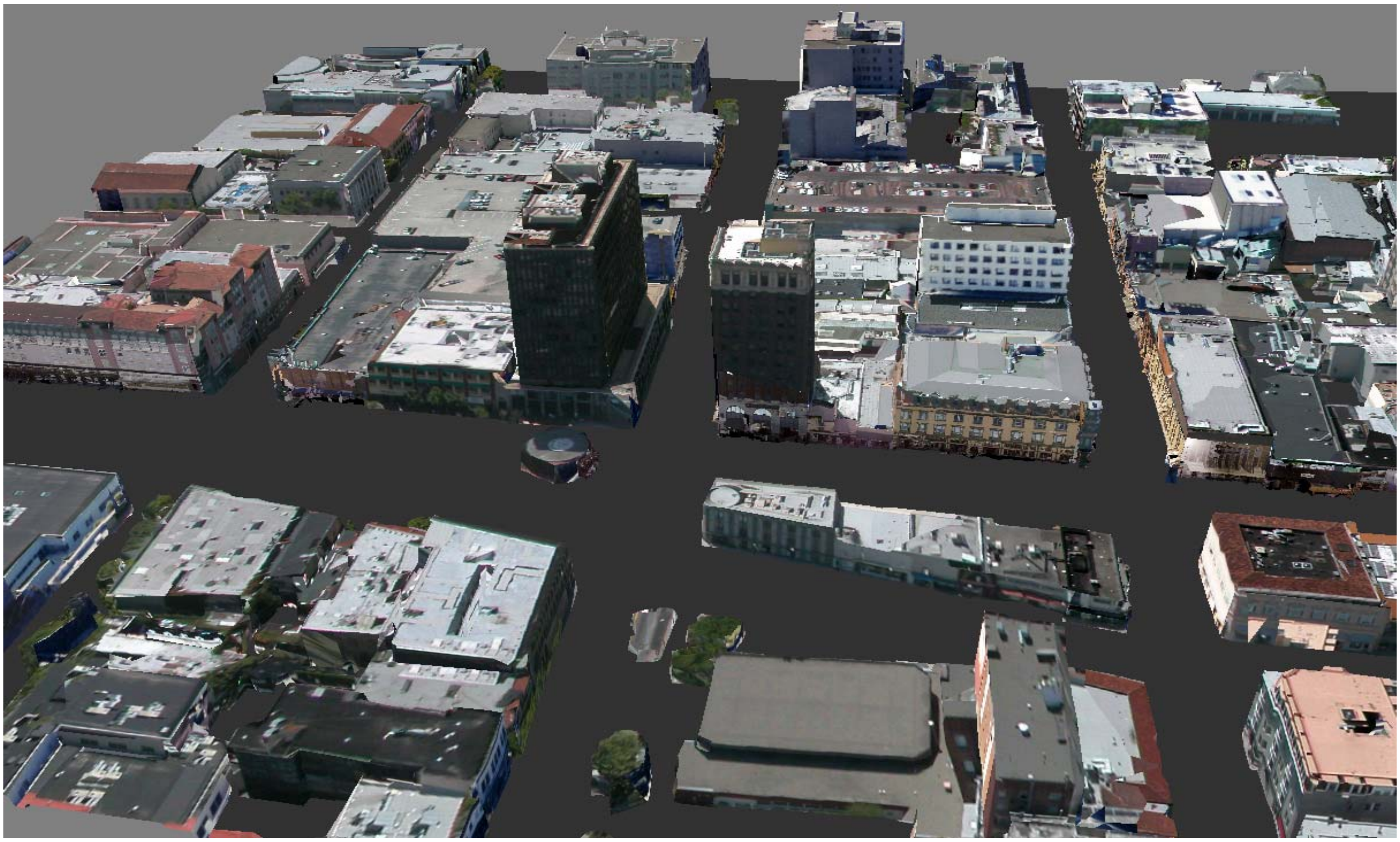


Figure 15.15: Bird's eye view of the model

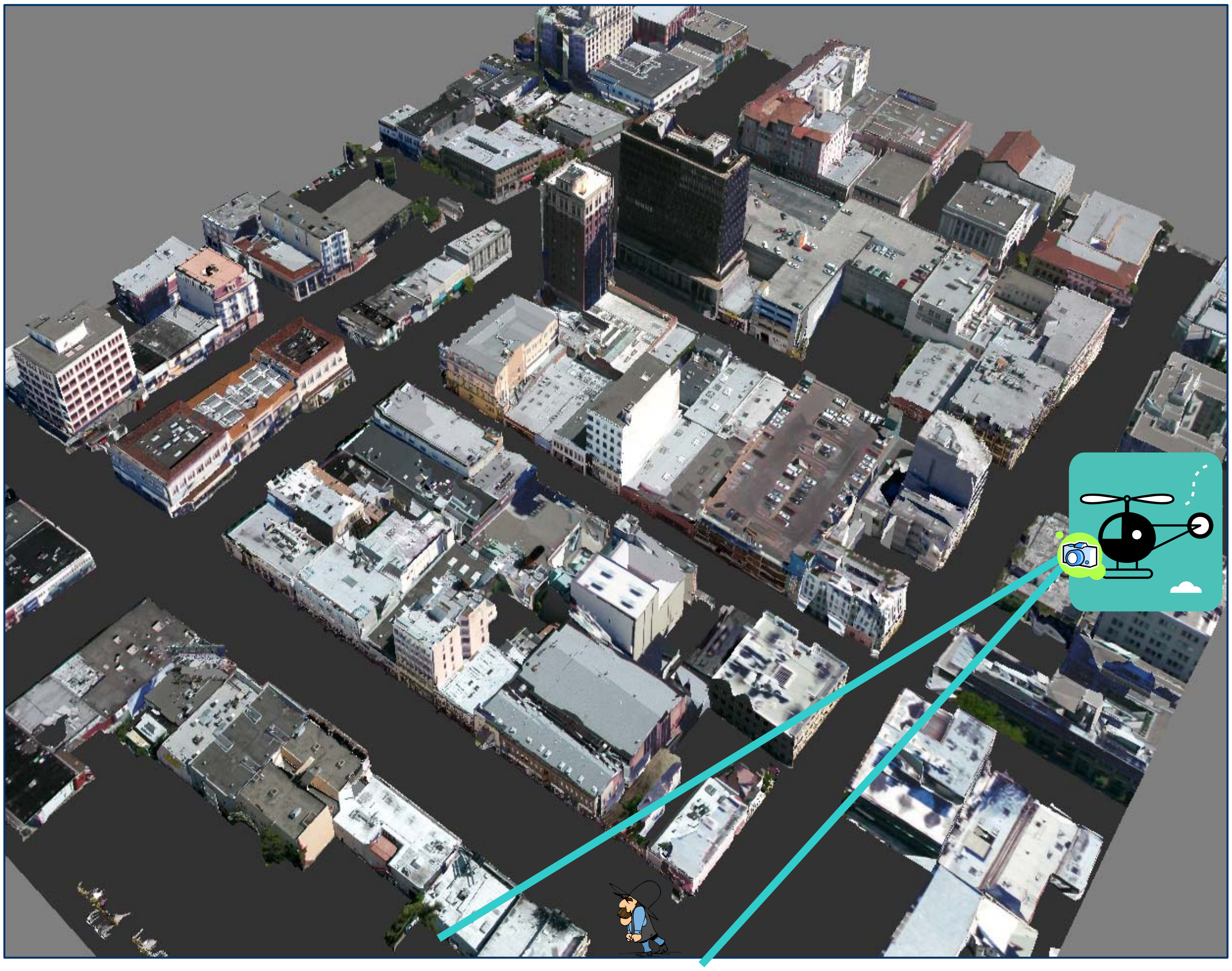


Figure 15.16: Illustration of ways in which a priori models of 3D cities can be used for surveillance applications.

Vehicle localization and registration with DSM	164 min
Facade model generation and optimization for rendering	121 min
DSM computation and projecting facade locations	8 min
Generating textured airborne mesh and blending	26 min
Total processing time	319 min

Table 15.1: Processing times for the downtown Berkeley blocks.