Constructing 3D City Models by Merging Ground-Based and Airborne Views

Christian Frueh and Avideh Zakhor[1]

Video and Image Processing Lab
University of California, Berkeley

In this paper, we present a fast approach to automated generation of textured 3D city models with both high details at ground level, and complete coverage for bird's-eye view. A close-range facade model is acquired at the ground level by driving a vehicle equipped with laser scanners and a digital camera under normal traffic conditions on public roads; a far-range Digital Surface Map (DSM), containing complementary roof and terrain shape, is created from airborne laser scans, then triangulated, and finally texture mapped with aerial imagery. The facade models are first registered with respect to the DSM by using Monte-Carlo-Localization, and then merged with the DSM by removing redundant parts and filling gaps. The developed algorithms are evaluated on a data set acquired in downtown Berkeley.

Keywords: localization, scan matching, airborne laser scans, 3D city model, urban simulation

I. Introduction

Three-dimensional models of urban environments are useful in a variety of applications such as urban planning, training and simulation for urban terrorism scenarios, 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 [5, 10, 15]. In recent years, advances in resolution and accuracy have rendered airborne laser scanners suitable for generating Digital Surface Maps (DSM) and 3D models [1, 9, 12] without error-prone camera parameter estimation, line or feature detection and 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 [1]. While sub-meter resolution can be achieved using this technique, only the roofs and not the facades of buildings are captured.

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. Most common approaches involve enormous amounts of manual work, such as importing the geometry obtained from construction plans; there have also been attempts to acquire this close-range data in an automated fashion, either using stereo vision [3] or 3D laser scanners [13]. 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 previous work [6, 7], we have developed an automated method capable of rapidly acquiring 3D geometry and texture data for an entire city at the ground level. We use a vehicle equipped with fast 2D laser scanners and a digital camera to acquire data to be processed offline, while driving at normal speeds on public roads. This approach has the advantage that data can be acquired continuously rather than in a stop-and-go fashion, and is therefore extremely fast. In [8], we have presented automated methods to process this data efficiently, in order to obtain a highly detailed model of the building facades in downtown Berkeley. These facade models however, do not provide any information about roofs or terrain shape - they are essentially the virtual equivalent of a Hollywood city.

In this paper, we will describe an approach to merging the highly detailed facade models with a complementary airborne model, in order to provide both the necessary level of detail for walk-thrus and the completeness for fly-thrus. We use a DSM from airborne laser scans to generate both an edge map for position estimation of the ground-based acquisition vehicle, and a large-scale, texture mapped aerial surface mesh. This airborne mesh is at one-meter resolution and provides a bird's-eye view over the entire area, as it contains terrain profile and building tops. In contrast to previous approaches, we do not explicitly extract a high-level polygonal model from the DSM. Rather, we merge the two different models with vastly different resolutions in order to obtain a complete model suitable for both walk- and fly-thrus.

The outline of this paper is as follows: Section II describes the generation of a DSM and a textured surface mesh from airborne laser scans. Section III details our approach to ground-based model generation and model registration. We propose a method to merge the two models in Section IV, and in Section V, we present results for a data set of downtown Berkeley.

II. Textured Surface Mesh from Airborne Laser Scans

In this section, we describe the generation of a DSM from airborne laser scans, its processing and transformation into a mesh, and texture mapping with a color aerial image. This DSM will be utilized for localizing the ground-based data acquisition vehicle, and for adding roofs and terrain to the ground-based facade models. While we use aerial laser scans to create the DSM, it is equally feasible to use other sources such as stereo vision or SAR.

A. Scan Point Resampling 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 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 scan points. 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 1.3 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 1 by 1 square meter, 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 side 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 intensity proportional to z, as shown in Figure 1(a).

B. Processing the DSM

The DSM 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 reconstruct a more visually pleasing reconstruction of the roofs, we apply additional processing steps:

The first step is aimed at flattening "bumpy" rooftops. To do this, we first apply to all non-ground pixels a region-growing segmentation algorithm 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 1(a) is shown in Figure 1(b).

The second processing step is intended to straighten jittery edges. We re-segment the DSM, detect the boundary points of each region, and use RANSAC to find line segments that approximate the regions. For the consensus computation, we also consider boundary points of surrounding regions, in order to detect even short linear sides of regions, and to align them consistently with surrounding buildings; furthermore, we reward an additional bonus consensus score 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 1(c) shows the regions resulting from processing Figure 1(b), superimposed with the corresponding RANSAC lines drawn in white. Compared with Figure 1(b), most edges are straightened out.

C. Creating Edge Map and DTM

In previous work [7], we have applied a Sobel edge detector to a grayscale aerial image in order to find edges in the city for localizing the ground-based data acquisition vehicle. For the DSM, rather than using the Sobel edge detector, we define a discontinuity detection filter, which marks a pixel if at least one of its neighboring 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, and not the 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 cause numerous false edges in the image, 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. 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. 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 Digital Terrain Map (DTM) as an estimate for the ground level near roads. Figures 1(d) and 1(e) show edge map and DTM, respectively, for the DSM shown in Figure 1(b).
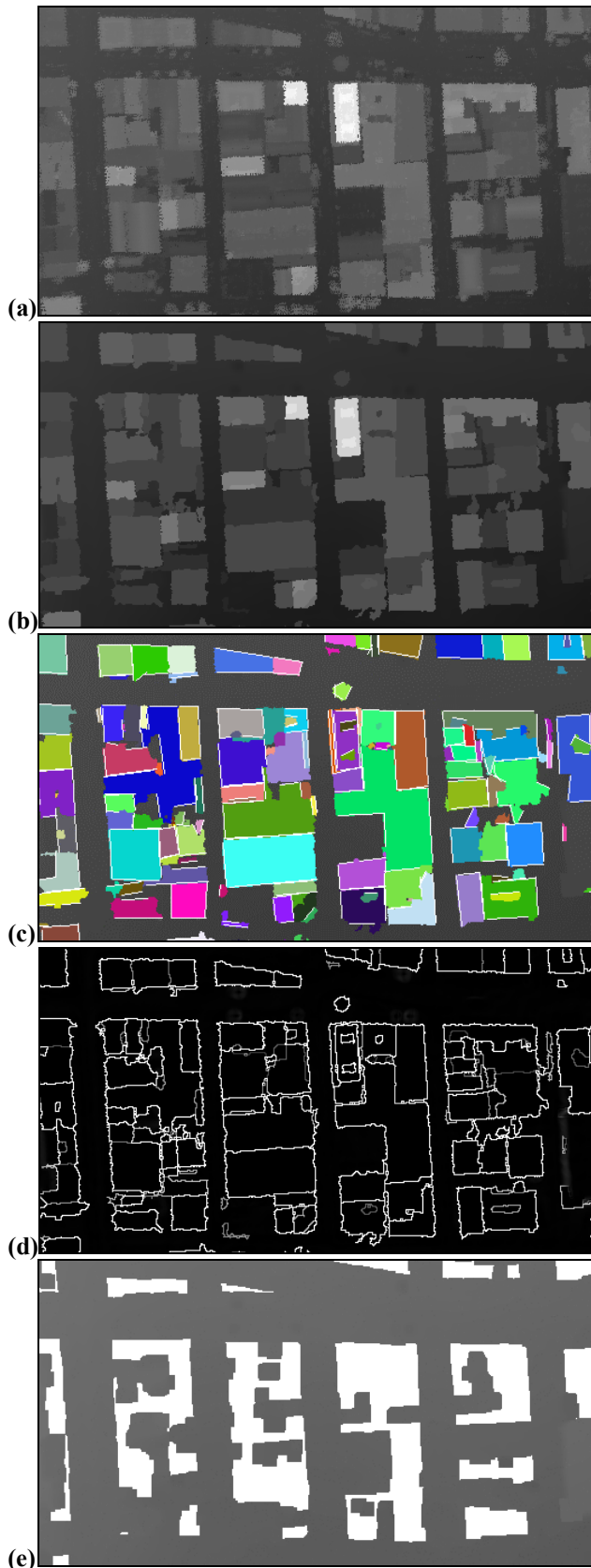
Figure 1: Processing steps for DSM; (a) DSM obtained from scan point resampling; (b) DSM after flattening roofs; (c) segments with white RANSAC lines; (d) edge map; (e) DTM. For the white pixels, there is no ground level estimate.

D. Textured Mesh Generation

Airborne models are commonly generated from LIDAR scans via a sophisticated process [1,12]. By detecting features such as planar surfaces in the DSM or matching a predefined set of possible rooftop and building shapes, a polygonal model of low-level 3D primitives can be found for the buildings. 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 - conditions that are particularly true of the Berkeley campus. Although the resulting models may appear "clean" and precise, the geometry and location of the reconstructed buildings is not necessarily correct if the underlying shape assumptions are invalid.

As we will describe in Section III, in our application, we have an accurate model of the building facades readily available, and are mainly interested in adding the complementary roof and terrain geometry. Since we do not face the complex problem of determining the correct footprint and facade height of the building sides facing the street, we can apply a simple technique to create a model from airborne views, namely transforming the DSM directly into a triangular mesh and reducing the number of triangles by simplification. The advantage of this method is its robustness and low processing complexity. Since no a priori assumptions about the environment are made or pre-defined models are required, this approach 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 and rather jittery edges for some locations where no facade model is available, and where edges have not been straightened in the DSM processing procedure.

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 two million triangles per square kilometer for the 1 meter by 1 m square 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, a reduction factor of 10 to 20 is possible without apparent loss in quality, resulting in a mesh with about 100,000 triangles per square kilometer at highest level-of-detail.

Using an aerial image taken from an unknown camera pose, the reduced mesh can be texture mapped in a semi-automatic way: correspondence points are selected manually in both the aerial photo and the DSM, taking only a few minutes for an entire city of several square kilometers. A location in the DSM corresponds to a 3D vertex in space and can be projected into the aerial image for a given camera pose. We utilize Lowe's algorithm to

compute the optimal camera pose by minimizing the difference between selected correspondence points and computed projections [11]. At least 6 correspondence points are necessary to solve for the pose, but in practice about 10 to 20 points are selected for robustness. After the camera pose is determined, the corresponding texture coordinate, i.e. image location, is computed for each vertex, and the mesh triangles are texture mapped accordingly. Figure 2 shows a surface mesh for the east side of the Berkeley campus. Since we do not make any distinction between various object types, the mesh contains the entire geometry, including buildings, trees, and terrain shape.
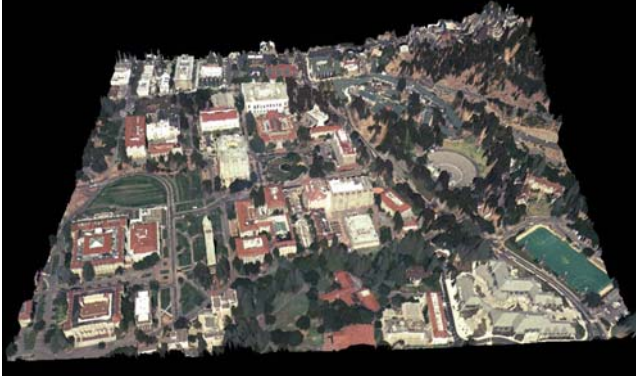

Figure 2: Texture mapped model of the east side of the Berkeley campus.

III. CLOSE-RANGE MODELING AND MODEL REGISTRATION

A. Ground-Based Data Acquisition

In previous work, we have developed a mobile data acquisition system consisting of two Sick LMS 2D laser scanners and a digital color camera with a wide-angle lens. This 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. Both 2D scanners face the same side of the street, one mounted horizontally, the other one vertically, and they are synchronized by hardware signals. 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. 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 reliable enough in urban canyons, we introduce in [7] the use of an aerial photo as a 2D global reference in conjunction with Monte-Carlo-Localization (MCL). In the following, we extend the application of MCL to a global edge map derived from the DSM, in order to determine the vehicle's pose, and to register the ground-based facade models with respect to the DSM.

B. Model Registration with Monte-Carlo-Localization

MCL is a particle-filtering-based implementation of the probabilistic Markov localization, and is used in mobile robotics to track the position of a vehicle. Given a series of relative motion estimates and corresponding horizontal laser scans, the MCL approach we have proposed in [7] is capable of determining the accurate position within a global edge map. We extend the 2D correction of this previous MCL approach by using the DSM rather than an aerial photo as a global reference; furthermore, we recover not only the 3 degrees of freedom (DOF) of the 2D pose, but also 5 of a full 6-DOF 3D pose.

The principle of the correction is to adjust initial vehicle motion estimates so that 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 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 to obtain an accurate localization of the acquisition vehicle.

Using the DTM computed in the previous section, an estimate for 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.

At the end of the above steps, we obtain 6-DOF estimates for the global pose, and can apply a framework of automated processing algorithms to remove foreground and reconstruct facade models. As described in [8], the path is segmented into easy-to-handle segments to be processed individually. The further steps include generation of a point cloud, classification of areas as facade versus foreground, removal of foreground geometry, filling facade holes, creating a mesh and texture mapping [8]. As a result, we obtain texture mapped facade models as seen in Figure 5. Note that the upper parts of tall buildings cannot be texture mapped if they have been outside the camera's field of view during data acquisition.

Figure 5: 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 bicubic 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 players.

IV. MODEL MERGING

In this section, we 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 potentially resulting in substantial discrepancies between the two meshes. Our goal is a photorealistic virtual exploration of the city, and hence creating models with visually pleasing appearances is more important than CAD properties such as watertightness. Common approaches for fusing meshes, such as sweeping and intersecting contained volume [2], 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 almost an order of magnitude higher than that of the airborne surface mesh. Furthermore, to enable interactive rendering, it is required that the two models fit together even when their parts are at different levels-of-detail.
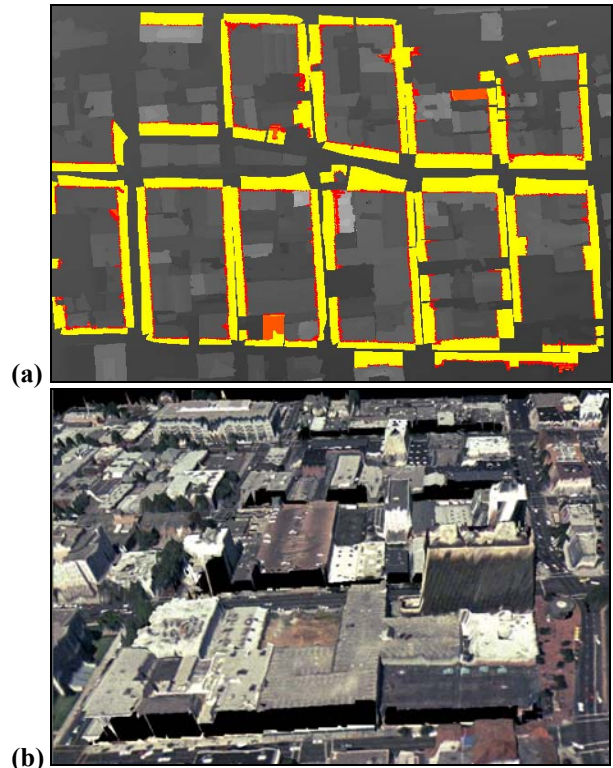

(a)

(b)

Figure 6: Removing facades from airborne model; (a) marked areas in DSM; (b) resulting mesh with corresponding facades and foreground objects removed.

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 localization techniques described earlier. We further identify and mark those areas, which our automated facade processing in [8] has classified as foreground. These marks control the subsequent airborne mesh generation from DSM; specifically, during the generation of the airborne mesh, (a) the z value for the foreground areas 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 6(a) shows the DSM with facade areas marked in red and foreground marked in yellow, and Figure 6(b) shows the resulting airborne surface mesh with

the corresponding facade triangles removed and the foreground areas leveled to DTM altitude.

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 7. Our approach to creating such a blend mesh is to extrude the buildings along an axis perpendicular to the facades, as shown in Figure 7(b), and then shift the location of the "loose end" vertices to connect to the closest airborne mesh surface, as shown in Figure 7(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.
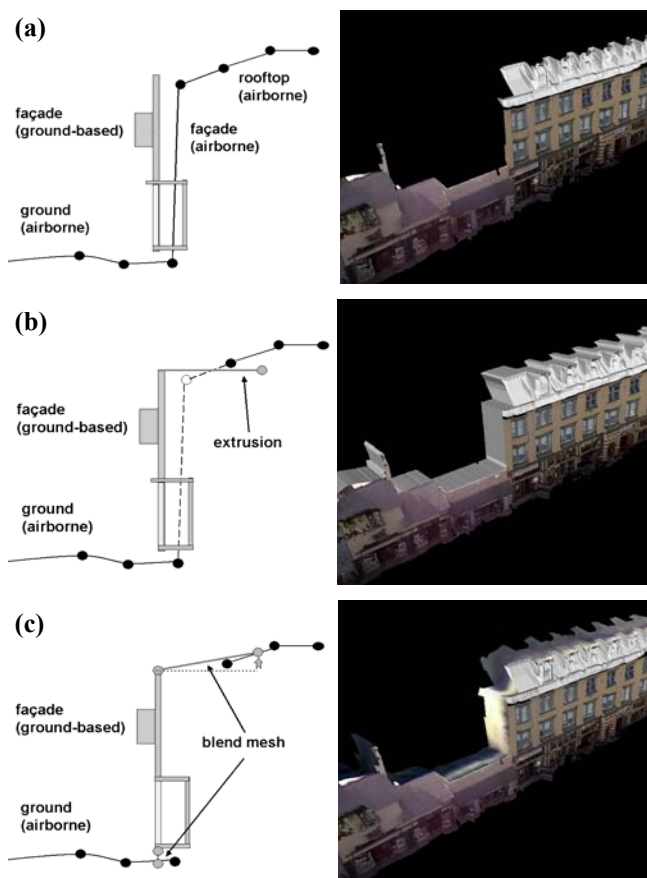

Figure 7: Creation of a blend mesh. A vertical cut through a building facade is shown. (a) Initial airborne and ground-based model registered; (b) facade of airborne model replaced and ground-based model extruded; (c) blending the two meshes by adjusting "loose ends" of extrusions to airborne mesh surface and mapping texture.

V. RESULTS

We have applied the proposed algorithms on a data set for downtown Berkeley. The airborne laser scans have been acquired in conjunction with Airborne 1, Inc. of Los Angeles, CA; the entire data set consists of 48 million scan points. We have selected a cell size of 1 m by 1 m square and have applied the resampling described in Section III to obtain a DSM, an edge map, and a DTM for the entire data set. The ground-based data has been acquired during two measurement drives in Berkeley: The first drive took 37 minutes 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 took 41 minutes and was 14.1 kilometers long, starting from Cory Hall 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.
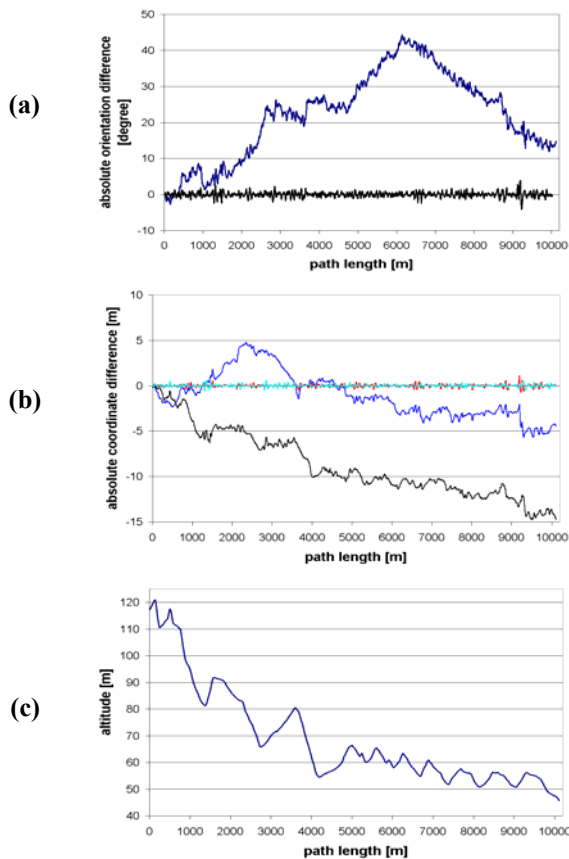

Figure 8: Global correction for path 1; (a) yaw angle difference between initial path and global estimates before and after correction; (b) differences of x and y coordinates before and after correction; (c) assigned z coordinates.

In previous MCL experiments based on edge maps from aerial images with 30 cm resolution, we have found the localization uncertainty to be enormous at some locations, due to false edges and perspective shifts; hence, in the past, we have had to use 120,000 particles during MCL in order

to approximate the spread-out probability distribution appropriately and track the vehicle reliably. For the edge map derived from airborne laser scans however, we have found that despite its lower resolution, the vehicle could be tracked with as few as 5000 particles. As shown in Figure 8 for path 1, we have applied the global correction first to the yaw angles as shown in Figure 8(a), then recomputed the path and applied the correction to the x and y coordinates, as shown in Figure 8(b). As expected, the global correction substantially modifies the initial pose estimates, thus reducing errors in subsequent processing. Figure 8(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 9(a) shows uncorrected paths 1 and 2 superimposed on the airborne DSM, Figure 9(b) shows the paths after global correction, and Figure 10 shows the ground based horizontal scan points for the corrected paths. As seen, path and horizontal scan points match the DSM closely after applying the global corrections.

Figure 10: Horizontal scan points for corrected paths.

After the Monte-Carlo-Localization and correction, all scans and images are geo-referenced. We generate a facade model for 12 street blocks of the downtown area using the processing steps described in [8]. Figure 11 shows the resulting facades; note that the acquisition time for the 12 downtown Berkeley blocks was only 25 minutes; this is the time that it took to drive the total of 8 kilometers around the blocks under city traffic conditions.

Figure 11: Reconstructed facade model for the downtown Berkeley area.

Figure 12: Bird's eye view of the combined model.

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
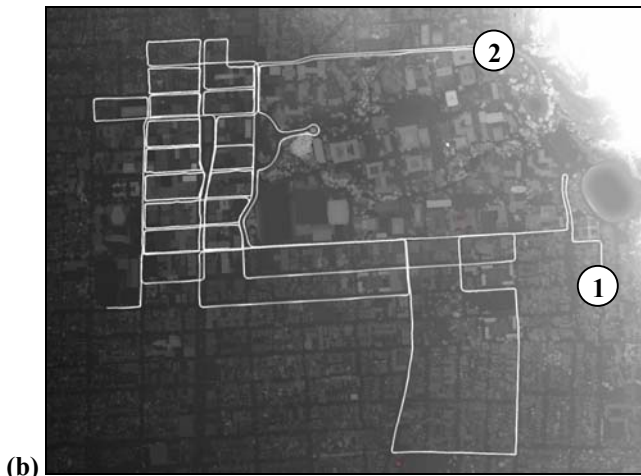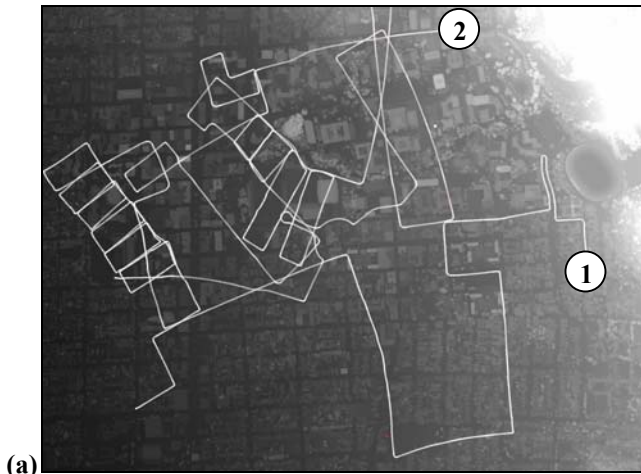

(a)

(b)

Figure 9: Driven paths superimposed on top of the DSM; (a) before correction, and (b) after correction. The round circles denote the starting position for path 1 and path 2, respectively.
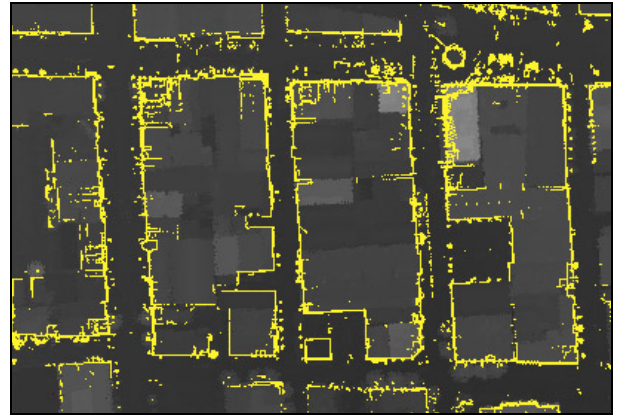
described in Section IV. Figure 12 shows the resulting combined model for the looped downtown Berkeley blocks from the top, as it appears in a fly-thru. Figure 13(a) shows the model as viewed in a walk-thru or drive-thru; in Figure 13(b), only the left side of the street has been replaced by the ground-based facade model, the right side is entirely derived from the airborne model for comparison purposes. As seen, the inserted ground-based facades are significantly more detailed and visually pleasing for walk-thru applications.
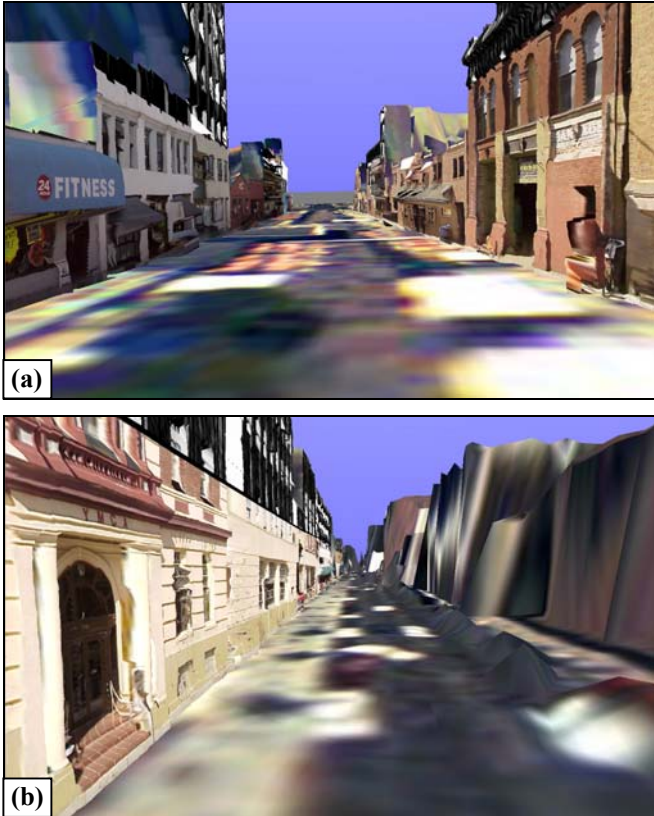

Figure 13: Walk-thru view of the combined model; (a) both sides of the street are from ground-based modeling; (b) only the facade on the left side of the street is from ground-based modeling for comparison purposes.

Vehicle localization and registration with DSM	164 min
Facade model generation and optimization for rendering	121 min
Selecting correspondences for registration of DSM and aerial image for texturing (manual, prorated to area)	2 min
DSM computation and projecting facade locations	8 min
Generating textured airborne mesh and blending	12 min
Total processing time	**307 min**

Table 1: Processing times for the 12 downtown Berkeley blocks.

Our proposed approach to city modeling is not only automated, but also fast from a computational viewpoint: As shown in Table 1, the total time for the automated processing and model generation for the 12 downtown blocks is around 5 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 and applicable to large environments.

VI. Conclusion and Future Work

We have presented a method of creating a 3D city model suitable for walk- and fly-thrus by merging models from airborne and ground-based views. Future work will address further improvements in the airborne mesh generation, adding 3D and 4D foreground components such as trees, traffic signs, cars and pedestrians to the ground-based model, and related rendering issues.

VII. References

[1]   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

[2]   B. Curless and M. Levoy, "A volumetric method for building complex models from range images", SIGGRAPH, New Orleans, 1996, pp. 303-312

[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, pp. 268-74

[4]   M. Garland and P. Heckbert, "Surface Simplification Using Quadric Error Metrics", SIGGRAPH '97, Los Angeles, 1997, pp. 209-216

[5]   D. Frere, J. Vandekerckhove, T. Moons, and L. Van Gool, "Automatic modeling and 3D reconstruction of urban buildings from aerial imagery", IEEE International Geoscience and Remote Sensing Symposium Proceedings, Seattle, 1998, pp. 2593-6

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

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

[8]   C. Frueh and A. Zakhor, "Data Processing Algorithms for Generating Textured 3D Building Facade Meshes From Laser Scans and Camera Images", 3D Processing, Visualization and Transmission 2002, Padua, Italy, 2002, pp. 834 - 847

[9]   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, pp.105-112

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

[11]   D. G. Lowe, "Fitting parmetrized three-dimensional models to images", Trans. On pattern analysis and machine intelligence, vol. 13, No. 5, 1991, pp. 441-450

[12]   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

[13]   I. Stamos and P.E. Allen, "3-D model construction using range and image data." CVPR 2000, Hilton Head Island, 2000, pp.531-6

[14]   S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping", ICRA 2000, San Francisco, 2000, vol. 1. 4, pp.321-8

[15]   C. Vestri and F. Devernay, "Using robust methods for automatic extraction of buildings", Computer Vision and Pattern Recognition, Hawaii, USA, 2001, vol.1. 2, pp. I-133-8