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

Christian Früh 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 Model (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 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, consisting of geometry and texture of visible surfaces, are useful in a variety of applications such as urban planning, training and simulation for disaster 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 [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]. 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. 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, either image-based [3] or by using 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 previous work [6, 7] we proposed an automated method capable of rapidly acquiring 3D geometry and texture data for an entire city at the ground level. This method uses a vehicle equipped with 2D laser scanners and a digital camera to acquire data to be processed offline, while driving at normal speeds on public roads. Zhao and Shibasaki [14] also proposed a similar system using 2D laser scanners and line cameras. In both systems, data is acquired continuously rather than in a stop-and-go fashion, and therefore the data acquisition process is extremely fast. In [8], we presented automated methods to process this type of data efficiently, in order to obtain a highly detailed model of the building facades in downtown Berkeley. However, these facade models do not provide any information about roofs or terrain shape, and only consist of surfaces visible from the ground level.

In this paper, we describe an approach to automatically register and merge our detailed facade models with a complementary airborne model, in order to provide both the necessary level of detail for walk-thrus, and the complete coverage for fly-thrus. The data flow diagram of our approach is shown in Figure 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]. Using the DSM obtained from airborne laser scans, we localize the acquisition vehicle, hence registering the ground-based facades to the airborne model, by means of Monte-Carlo-Localization (MCL). We merge the two models with vastly different resolutions in order to obtain a 3D model suitable for both walk- and fly-thrus. The remainder of this paper provides details on various components of Figure 1. More details on our approach are included in [6-8].

[1] This work was sponsored by Army Research Office contract DAAD19-00-1-0352.

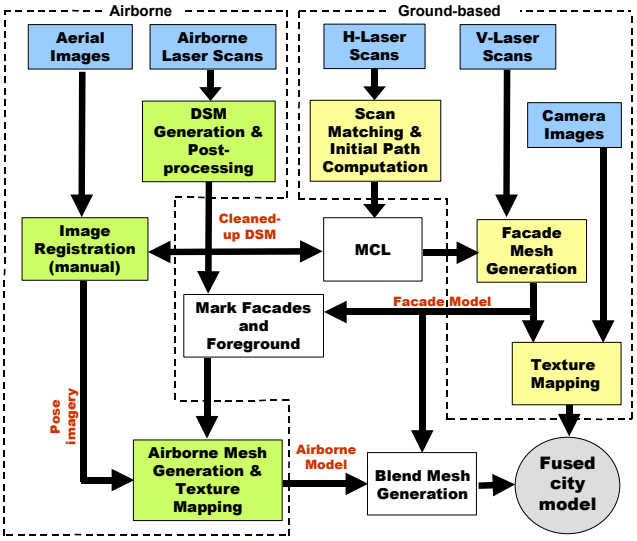

Figure 1: Data flow diagram of our modeling approach. Airborne modeling steps are highlighted in green, ground-based modeling steps in yellow, and model fusion steps in white.

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 outlines our approach to ground-based model generation and model registration. We propose a method to fuse 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 surface mesh, and texture-mapping with color aerial imagery. The DSM will be utilized for localizing the ground-based data acquisition vehicle, and for adding roofs and terrain to the ground-based facade models; in contrast to previous approaches, we do not explicitly extract geometric primitives from the DSM. 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.

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 point 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 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 intensity proportional to z.

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 obtain a more visually pleasing reconstruction of the roofs, we apply several 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 2(a) is shown in Figure 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 even 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 2(c) shows the regions resulting from processing Figure 2(b), superimposed with the corresponding RANSAC lines drawn in white. Compared with Figure 2(b), most edges are straightened out.

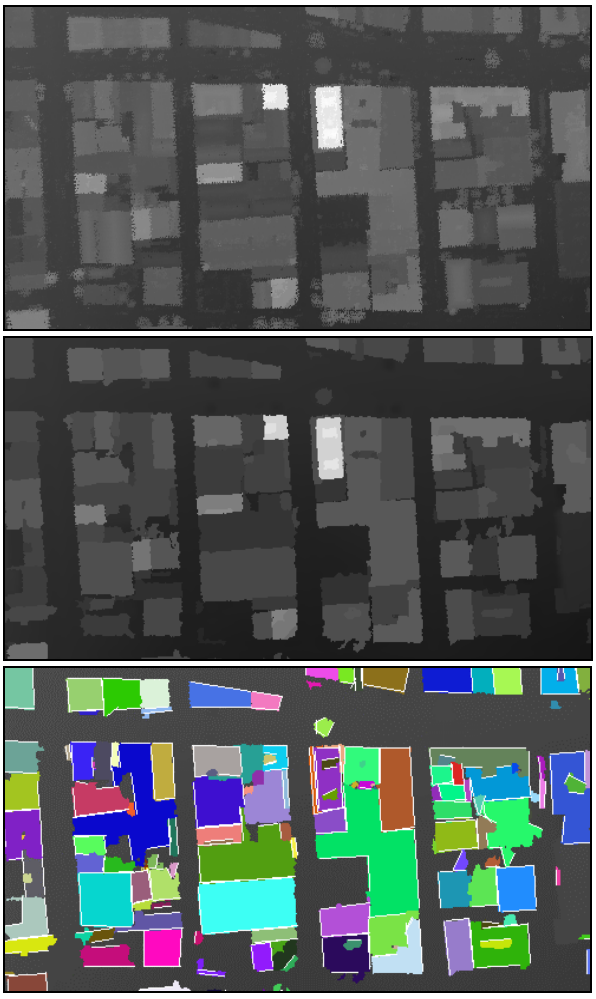

Figure 2: Processing steps for DSM; (a) DSM obtained from scan point resampling; (b) DSM after flattening roofs; (c) segments with RANSAC lines in white.

C. 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]. 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 - 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, 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 IV. 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 uncalibrated 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.

III. Ground-Based Modeling and Model Registration

A. *Ground-Based Data Acquisition via Drive-by Scanning*

In previous work, we have developed a mobile ground-based data acquisition system consisting 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 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 3: Acquisition vehicle.

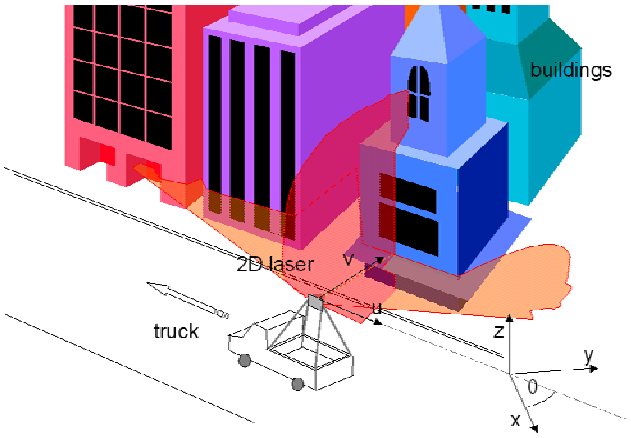

Figure 4: Ground-based acquisition setup.

Both 2D scanners face the same side of the street; one is mounted horizontally, the other vertically, as shown in Figure 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 the following, we extend the application of MCL to a global edge map 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.

B. *Creating Edge Map and DTM*

For the MCL approach described in the Section IIIc, we need to create two additional maps: an edge map, which contains the location of height discontinuities, and a Digital Terrain Model (DTM), 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 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 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, 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 DTM as an estimate of the ground level near roads. Figure 5(a) and (b) show edge map and DTM, respectively, computed from the DSM shown in Figure 2(b).

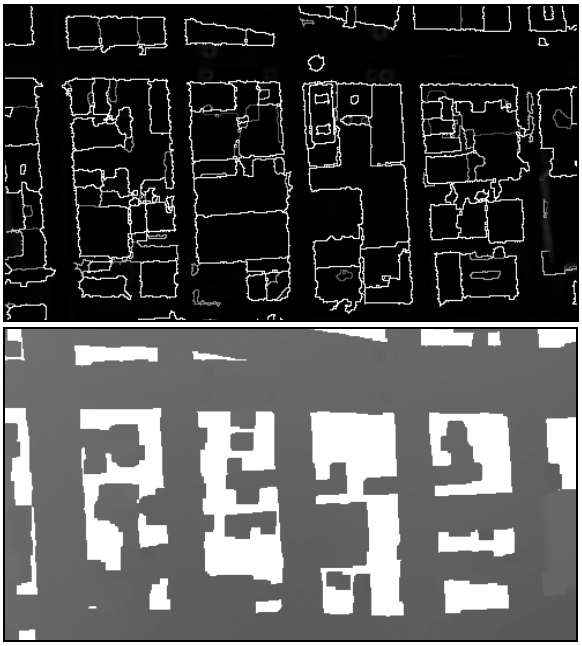

Figure 5: Map generation for MCL; (a) edge map; (b) DTM. For the white pixels, there is no ground level estimate available.

C. 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. Given a series of relative motion estimates and corresponding horizontal laser scans, the MCL-based approach we have proposed in [7] is capable of determining the accurate position within a global edge map. 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, 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.

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 and windows, creation of a surface mesh and texture-mapping [8]. As a result, we obtain texture-mapped facade models as shown in Figure 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.

Figure 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 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 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 [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 localization techniques described earlier. We further identify and mark those areas, which our automated facade processing in [8] has classified as foreground such as trees and cars. 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 7(a) shows the DSM with facade areas marked in red and foreground marked in yellow, and Figure 7(b) shows the resulting airborne surface mesh with the corresponding façade triangles removed, and the foreground areas leveled to DTM altitude.

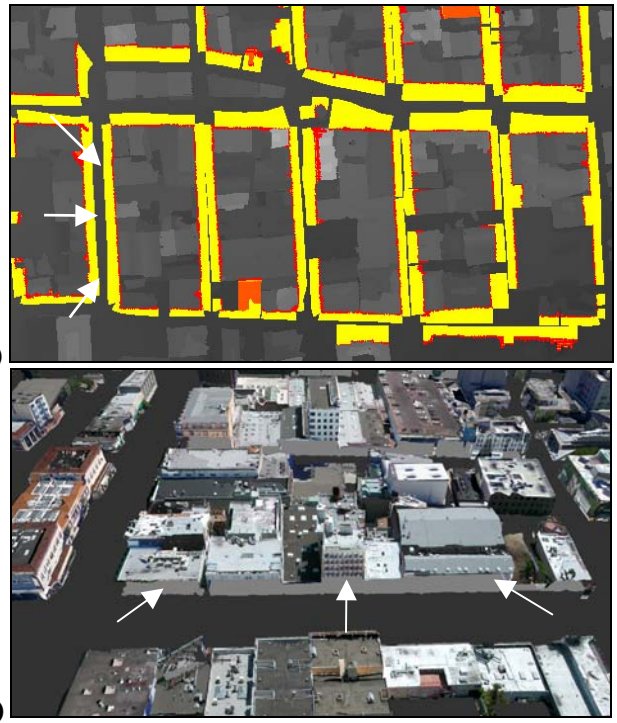

Figure 7: Removing facades from the airborne model; (a) marked areas in the DSM; (b) resulting mesh with corresponding facades and foreground objects removed. The arrows in (a) and in (b) mark corresponding locations in DSM and mesh, respectively.

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 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 8(b), and then shift the location of the "loose end" vertices to connect to the closest airborne mesh surface, as shown in Figure 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.

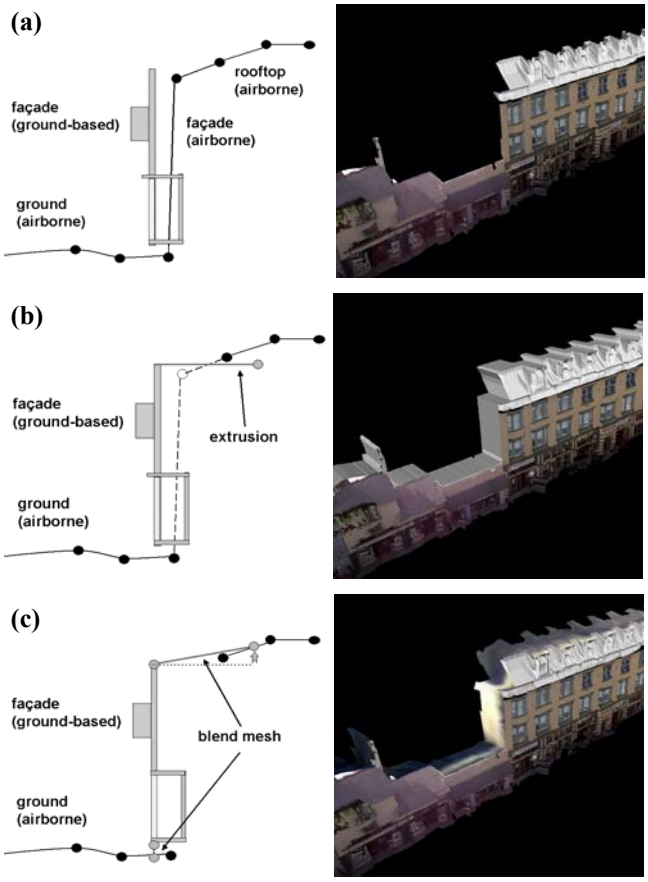

Figure 8: Creation of a blend mesh. A vertical cut through a building facade is shown. (a) Initial airborne and ground-based model registered; (b) facade of airborne model replaced and ground-based model extruded; (c) blending the two meshes by adjusting "loose ends" of extrusions to airborne mesh surface and mapping texture.

V. 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 resampled these scan points to a 0.5 m × 0.5 m grid, and have applied the processing steps as described in Section III to obtain a DSM, an edge map, and a DTM for the entire area. We select feature points in five-megapixel 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 DTM is automatically triangulated, simplified, and finally texture-mapped. Figure 9(a) shows the surface mesh obtained from directly triangulating the DSM, 9(b) shows the triangulated DSM after the processing steps, and 9(c) shows the texture-mapped model. 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. However, 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.

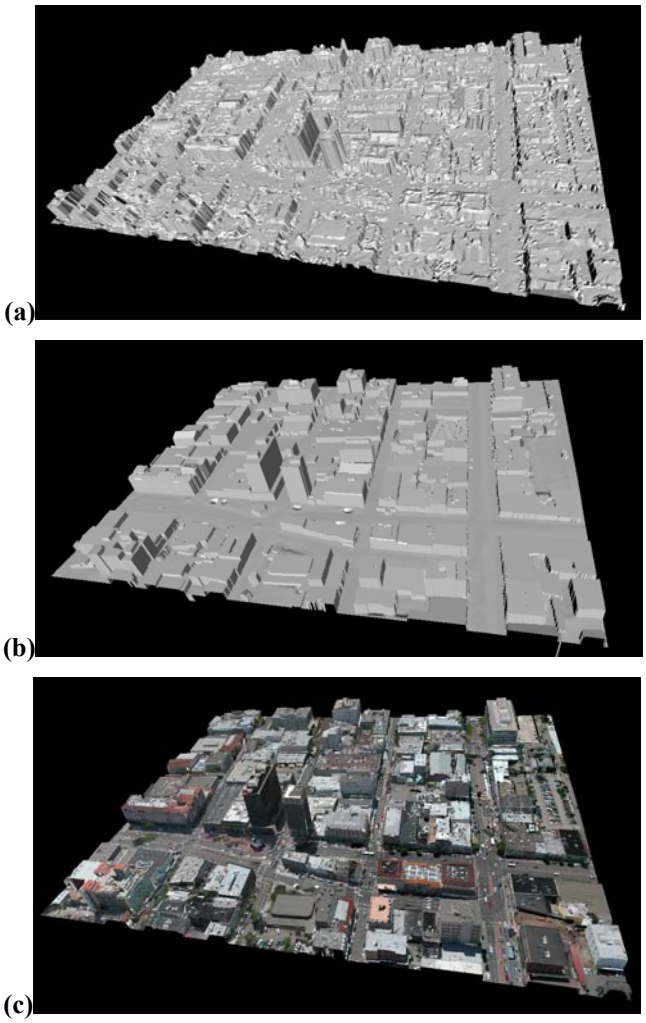

Figure 9: Airborne model. (a) DSM directly triangulated, (b) triangulated after postprocessing, (c) model texture-mapped.

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 was 41 minutes and 14.1 kilometers, starting from Cory Hall at U.C. Berkeley and looping around the remaining downtown blocks. A total of 332575 vertical and horizontal scans, consisting of 85 million scan points, along with 19200 images, were captured during those two drives.

In previous MCL experiments based on edge maps from aerial images with 30 cm resolution, we had 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 10 for path 1, we have applied the global correction first to the yaw angles as shown in Figure 10(a), then recomputed the path and applied the correction to the x and y coordinates, as shown in Figure 10(b). As expected, the global correction substantially modifies the initial pose estimates, thus reducing errors in subsequent processing. Figure 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.

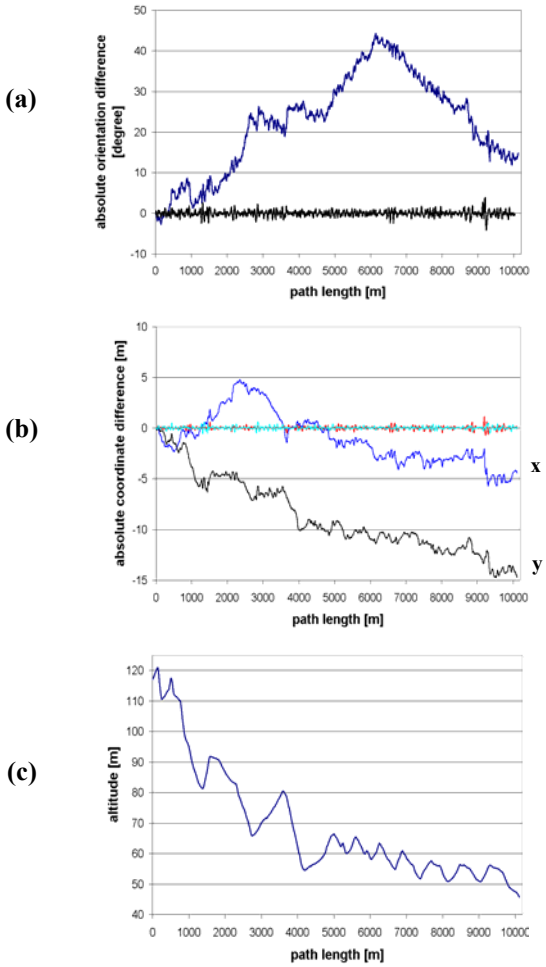

Figure 10: Global correction for path 1; (a) yaw angle difference between initial path and global estimates before and after correction; (b) differences of x and y coordinates before and after correction; (c) assigned z coordinates. In plots (a) and (b), the differences after corrections are the curves close to the horizontal axis.

Figure 11(a) shows uncorrected paths 1 and 2 superimposed on the airborne DSM, Figure 11(b) shows the paths after global correction, and Figure 12 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.

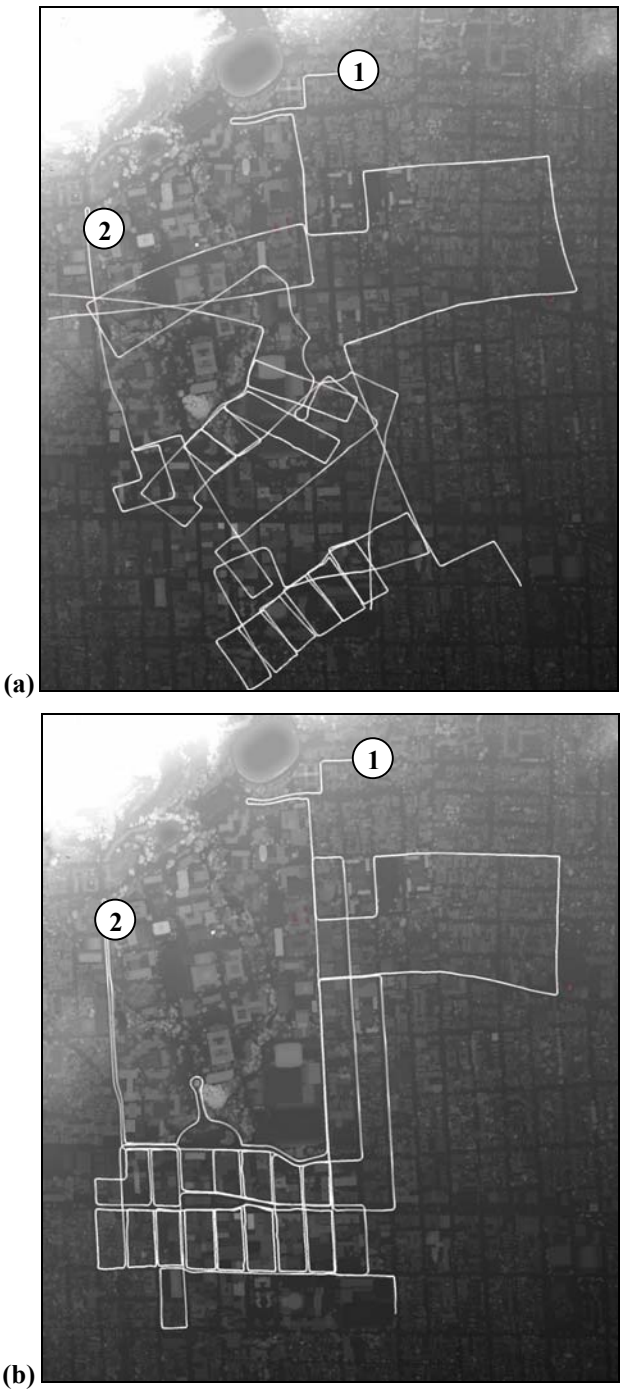

Figure 11: Driven paths superimposed on top of the DSM (a) before correction, and (b) after correction. The circles denote the starting position for paths 1 and 2, respectively.

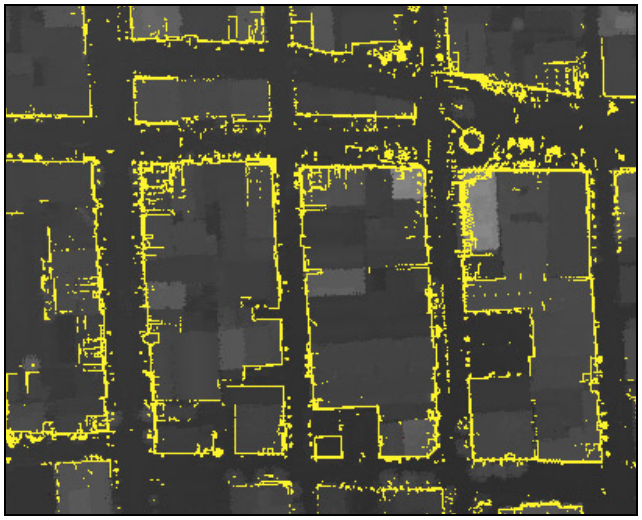

Figure 12: Horizontal scan points for corrected paths.

After MCL correction, all scans and images are geo-referenced. We generate a facade model for the 12 street blocks of the downtown area using the processing steps described in [8]. Figure 13 shows the resulting facades; note that the acquisition time for the 12 downtown Berkeley blocks has been only 25 minutes; this is the time portion of both paths that it took to drive the total of 8 kilometers around these 12 blocks under city traffic conditions.

Figure 13: Facade model for the 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 14(a) shows the resulting combined model for the looped downtown Berkeley blocks, as viewed in a walk-thru or drive-thru, and Figure 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 14(b) emphasizes the necessity of ground-based facade models for walk-thru applications.

Figure 15 shows the same model 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 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: 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 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.

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 1: Processing times for the downtown Berkeley blocks.

VI. Discussion and Future Work

We have presented an automated, fast method for creating 3D city models suitable for walk- and fly-thrus by merging models from airborne and ground-based views. While we have shown that our approach results in visually acceptable models for downtown environments, one of its limitations has to do with the way it handles foreground objects such as cars and trees. In particular, we currently remove such objects in order to avoid the difficult problem of reconstructing and rendering them. If too many trees are present, e.g. in residential areas, our underlying assumptions of dominant building planes are often not met, and hole filling entirely occluded facades is no longer reliable; therefore, in residential areas, the resulting model contains some artifacts. Furthermore, it is desirable to include common city objects such as cars, street lights, signs and telephone lines, for they substantially contribute to a high level of photo realism. Hence, future work will address reconstructing and adding 3D and 4D foreground components, e.g. by utilizing multiple scanners at different oblique directions.

Manually selecting correspondence points for registering the aerial imagery is the only manual step in our entire processing chain, and thus it is desirable to automate this process as well. This could be solved by utilizing an accurate GPS/INS unit, or by applying model-based vision methods such as finding vanishing points or matching features in DSM and images. Finally, the high level of detail of our method results in enormous amounts of data, and for many applications, a compact representation is desirable. Furthermore, the large data size, in particular the amount of high-resolution texture, makes rendering a challenging task, and future work has to address data management issues related to rendering models that are many orders of magnitude larger than the memory of existing computer systems.

VII. 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. 2

[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/