International Journal of Computer Vision 60(1), 5–24, 2004
© 2004 Kluwer Academic Publishers. Manufactured in The Netherlands.

An Automated Method for Large-Scale, Ground-Based City Model Acquisition

CHRISTIAN FRÜH AND AVIDEH ZAKHOR
Video and Image Processing Laboratory, University of California, Berkeley
frueh@eecs.berkeley.edu
avz@eecs.berkeley.edu

Received December 27, 2002; Revised December 11, 2003; Accepted December 11, 2003

Abstract. In this paper, we describe an automated method for fast, ground-based acquisition of large-scale 3D city models. Our experimental set up consists of a truck equipped with one camera and two fast, inexpensive 2D laser scanners, being driven on city streets under normal traffic conditions. One scanner is mounted vertically to capture building facades, and the other one is mounted horizontally. Successive horizontal scans are matched with each other in order to determine an estimate of the vehicle's motion, and relative motion estimates are concatenated to form an initial path. Assuming that features such as buildings are visible from both ground-based and airborne view, this initial path is globally corrected by Monte-Carlo Localization techniques. Specifically, the final global pose is obtained by utilizing an aerial photograph or a Digital Surface Model as a global map, to which the ground-based horizontal laser scans are matched. A fairly accurate, textured 3D cof the downtown Berkeley area has been acquired in a matter of minutes, limited only by traffic conditions during the data acquisition phase. Subsequent automated processing time to accurately localize the acquisition vehicle is 235 minutes for a 37 minutes or 10.2 km drive, i.e. 23 minutes per kilometer.

Keywords: laser scanning, navigation, self-localization, mobile robots, 3D modeling, Monte-Carlo localization

1. Introduction

Three-dimensional models of urban environments are used in applications such as urban planning, virtual reality, and propagation simulation of radio waves for the cell phone industry. Currently, acquisition of 3D city models is difficult and time consuming. Existing large-scale models typically take months to create, and usually require significant manual intervention (Chan et al., 1998). This process is not only prohibitively expensive, but also is unsuitable in applications where a 3D snapshot of a city is needed within a short time, e.g. for disaster management or for monitoring changes over time.

There exist a variety of approaches to creating 3D models of cities from an airborne view via remote sensing. Manual or automated matching of stereo images can be used to obtain a Digital Surface Model (DSM), i.e. a grid-like representation of the elevation level, and 3D models (Frere et al., 1998; Huertas et al., 1999). In recent years, advances in resolution and accuracy of Synthetic Aperture Radar (SAR) and airborne laser scanners have also rendered them suitable for the generation of DSMs and 3D models (Brenner et al., 2001; Maas, 2001). Although these methods can be reasonably fast, the resulting resolution of the models is only in the meter range, and without manual intervention, the resulting accuracy is often poor. Specifically, they lack the level of detail that is required for realistic virtual walk-throughs or drive-throughs.

While acquiring detailed building models from a ground-level view has been addressed in previous work, these attempts have been limited to one or a few buildings. Debevec et al. (1996) proposes to reconstruct

buildings based on few camera images in a semi-automated way. Similarly, it is conceivable to apply other vision-based approaches mainly designed for indoor scenes to outdoor environments, such as structure-from-motion methods (Koch et al., 1999) or variations (Dellaert et al., 2000; Szeliski and Kang, 1995), or voxel-based approaches (Seitz and Dyer, 1997), but varying lighting conditions, the scale of the environment, and the complexity of outdoor scenes with many trees and glass surfaces pose enormous challenges to purely vision-based methods.

Stamos and Allen (2000) use a 3D laser scanner and Thrun et al. (2000) and Hähnel et al. (2001) use 2D laser scanners mounted on a mobile robot to achieve complete automation, but the time required for data acquisition of an entire city is prohibitively large; in addition, the reliability of autonomous mobile robots in outdoor environments is a critical issue. Antone and Teller (2000) propose an approach based on high-resolution half-spherical images, but data has to be acquired in a stop-and-go fashion. While pose computation is sophisticated in this approach, the purely image-based model reconstruction is difficult and the obtained models are rather simple. Kawasaki et al. (1999) suggest a method, which uses a video stream for texturing an already existing 3D model. Zhao and Shibasaki (1999) use a vertical laser scanner in a car–based approach. While this enables the traversal of large-scale city environments in reasonable time, their localization is based on the Global Positioning System (GPS). GPS is by far the most common source of global position estimates in outdoor environments; however, it has several drawbacks: First, GPS tends to fail in dense urban environments, particularly in urban canyons where few satellites are visible. Second, multi-path reflections can result in completely erroneous readouts, or can decrease accuracy substantially. Third, a differential GPS system that could fulfill the hard accuracy requirements of ground-based modeling is quite expensive.

On the other hand, digital roadmaps and perspective-corrected aerial photos are widely available; similarly, for an increasing number of urban areas, DSMs can be found. Since both photos and DSM can provide a geometrically correct view of an entire city, it is conceivable to use them as a global map in order to arrive at global position without use of GPS devices. Another advantage of using an aerial photo or a DSM over GPS is that the airborne data can potentially be used to derive 3D models of a city from a bird's eye view, which can then be merged with the 3D facade models obtained from ground level laser scans.

In this paper, we propose "drive-by scanning" as a method that is capable of rapidly acquiring 3D geometry and texture data of an entire city at the ground level by using a configuration of two fast, inexpensive 2D laser scanners and a digital camera. This data acquisition system is mounted on a truck moving at normal speed on public roads, collecting data to be processed offline. This approach has the advantage that data can be acquired continuously, rather than in a stop-and-go fashion, and is therefore much faster than existing methods based on 3D scanners. While 3D scans overlap and can hence be accurately registered with each other if an initial pose is known approximately, this is not possible in our approach, since vertical 2D scans are parallel and do not overlap, hence posing more stringent accuracy requirements for the localization of the vehicle and its acquisition devices. More specifically, it is necessary to determine the pose of successive laser scans and camera images in a global coordinate system with centimeter and sub-degree accuracy in order to reconstruct a consistent model.

In our approach, rather than GPS, we use an aerial image or an airborne DSM to precisely reconstruct the path of the acquisition vehicle in offline computations: First, relative position changes are computed with centimeter accuracy by matching successive horizontal laser scans against each other, and are concatenated to reconstruct an initial path estimate. Since small errors and occasional mismatches can accumulate to a significantly large level after longer driving periods, Monte-Carlo-Localization (MCL) is then used in conjunction with an airborne map to correct global pose. Our approach is to match features observed in both the laser scans and the aerial image or the DSM, whereby the airborne data can be regarded as a global map onto which the ground-based scan points have to be registered.

This problem is in many ways similar to the localization of mobile robots, and as such, several approaches have been developed in this context: Cox (1991) proposed to match scan points from a laser scanner with the lines of a manually created a-priori-map of an indoor environment. Lu and Milios (1994) proposed the iterative dual correspondence or IDC algorithm, which is based on the matching-range-point rule. Gutmann and Schlegel (1996) focused on matching two scans, and proposed the use of a line filter for both

reference and second scans. There are several probabilistic attempts to localize a robot in a given edge map of the environment: Jensfelt and Kristensen (1999) and Roumeliotis and Bekey (2000) propose multi-hypotheses Kalman filters, which represent the pose belief as mixtures of Gaussians. Markov Localization, which assumes static environments, has been applied in Russell and Norvig (1995), Simmons and Koenig (1995) and Fox et al. (1999). In grid-based Markov localization, the parameter space is partitioned into grid cells, each representing the probability in a parameter "cube" by a floating point value, e.g. in Burgard (1996) and Fox et al. (1999). In MCL, also known as particle filtering or as condensation algorithm, a large number of random samples, or particles, is utilized to represent probability distributions (Fox et al., 2000; Thrun et al., 2001).

Lu and Milios (1997b), Thrun et al. (1998b) and Gutmann and Konolige (1999) have investigated simultaneous map building and localization in indoor environments by establishing cross-consistency over multiple 2D laser scans, without the use of a global map. However, these methods are not applicable to outdoor scale, since their complexity usually increases as $O(n^2)$ with n denoting the number of scans. Additionally, cities are extremely cyclic environments, with often no cross-correspondences to other scans during long driving periods. What makes our problem different from indoor localization is the scale of the environment, because distances involved in making 3D models for cities are large compared to the range of the laser scans. Furthermore, our approach is to obtain relative motion not from odometry, but from scan-to-scan matching, and the global map derived from a photo or a DSM does not have the same quality as a CAD ground plan of an indoor environment. Finally, while indoor localization is usually a 2D problem, in this paper, we recover a 6-degree-of-freedom (DOF) pose in case of a DSM as a global map.

The outline of this paper is as follows: Section 2 describes the system overview and Section 3 the data acquisition system. Section 4 is devoted to the relative pose estimation and initial path computation based on laser scan matching. In Section 5, we address global map generation from an aerial image or a DSM, and in Section 6, we discuss pose correction based on registration of laser scans with the map by means of MCL. Section 7 briefly outlines the model generation, and we finally show the results for a 10-kilometer drive in an actual city environment in Section 8.

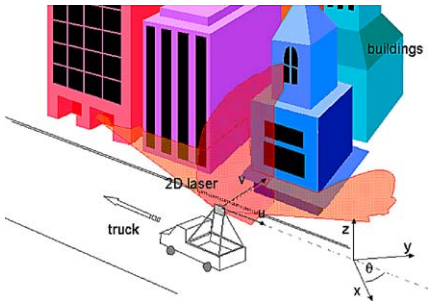

Figure 1. Experimental setup.

2. System Overview

The data acquisition system is mounted on a truck and consists of two parts: a sensor module and a processing unit (Früh and Zakhor, 2001a). The processing unit consists of a dual processor PC, large hard disk drives, and additional electronics for power supply and signal shaping; the sensor module consists of two SICK 2D laser scanners, a digital camera, and a heading sensor. It is mounted on a rack at a height of approximately 3.6 meters, in order to avoid moving obstacles such as cars and pedestrians in the direct view. The scanners have a 180° field of view with a resolution of 1°, a range of 80 meters and an accuracy of ±3.5 centimeters. Both 2D scanners face the same side of the street. One is mounted vertically with the scanning plane orthogonal to the driving direction, and the other is mounted horizontally with the scanning plane parallel to the ground. Figure 1 shows the experimental setup for our data acquisition.

The vertical scanner detects the shape of the building facades as we drive by, and therefore we refer to our method as drive-by scanning; the horizontal scanner operates in a plane parallel to the ground and is used for pose estimation as described in this paper. The camera's line of sight is the intersection between the orthogonal scanning planes. All sensors are synchronized with each other and acquire data at prespecified times. Figure 2 shows a picture of the truck with rack and equipment.

3. Relative Position Estimation and Path Computation

In this section, we compute relative pose estimates and an initial path by matching successive horizontal laser scans. A popular approach for matching two 3D scans

Figure 2. Truck with acquisition equipment.

is the Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992), and its 2D equivalent Iterative Dual Correspondence (IDC) (Lu and Milios, 1994). Both algorithms work directly on the scan points rather than on extracted features, and iteratively refine pose and point-to-point correspondences in order to converge to a final pose estimate. Our scan matching approach is similar to the line-based ones in Cox (1991) and Gutmann and Schlegel (1996), except for two modifications: First, we use both lines and single points of the reference scan for matching; second, we do not treat the problem of eliminating erroneous correspondences separately from the matching; rather, we consider it directly in the computation of a match quality function by using robust least squares, as will be seen later.

We first introduce a Cartesian world coordinate system $[x, y, z]$ where x, y defines the geographical ground plane and z the altitude as shown in Fig. 1. We also define a truck coordinate system $[u, v]$ which is implied by the horizontal laser scanner. In a flat environment, a 2D pose of the truck and its local coordinate system can be entirely described by the two coordinates x, y and the yaw orientation angle θ; in this case, the u-v coordinate system is assumed to be parallel to the x-y plane as shown in Fig. 1.

Since horizontal scans are taken continuously during driving and hence overlap substantially, the relative pose between the two capture positions can be determined by matching their corresponding laser scans, as shown in Fig. 3. We assume that the two scans have maximum congruence for the particular translation $\vec{t} = (\Delta u, \Delta v)$ and rotation $\Delta\varphi$, which exactly compensate for the motion between the two acquisition times. In practice however, scans taken from different positions do not match perfectly because of different sample density on objects, occlusions, and measurement noise. We will address the first issue by linear interpolation between scan points, and the two others by utilizing robust least squares as an outlier-tolerant way to find the pose with the smallest possible discrepancy.

Taking one scan as reference scan, we maximize $Q = f(\Delta u, \Delta v, \Delta\varphi)$, which computes the quality of alignment as a function of a given displacement $\Delta u, \Delta v$ and rotation $\Delta\varphi$ of the scans against each other. More specifically, we perform the following steps: First, we compute a set of lines l_i from the reference scan by connecting adjacent scan points provided their discontinuity is below a threshold; this results in a line strip approximation that may also contain isolated points as infinitely short lines. In this fashion, the reference scan is transformed into an edge map to which the second scan can be registered.

Given a translation vector $\vec{t} = (\Delta u, \Delta v)$ and a 2×2 rotation matrix $R(\Delta\varphi)$ with rotation angle $\Delta\varphi$, we

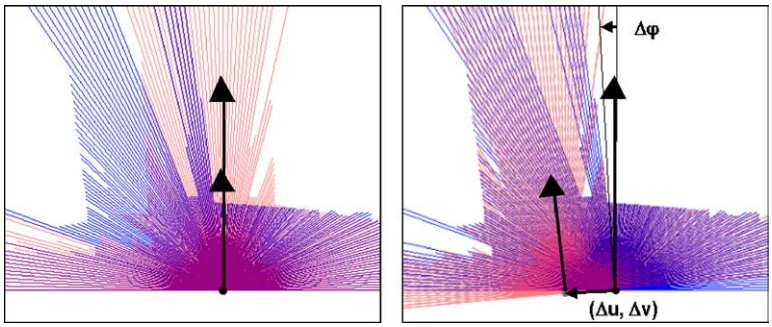

Figure 3. Two scans before matching (left) and after matching (right).

transform the points p_j of the second scan to the points p'_j according to

$$\vec{p}'_j = (\Delta u, \Delta v, \Delta\varphi) = R(\Delta\varphi) \cdot \vec{p}_j + \vec{t} \qquad (1)$$

Then, for each point $\vec{p}'_j$, we compute the Euclidean distance $d(\vec{p}'_j, l_i)$ to each line segment l_i and set $d_{\min}$ to:

$$d_{\min}(\vec{p}'_j(\Delta u, \Delta v, \Delta\varphi)) = \min_i\{d(\vec{p}'_j, l_i)\}. \qquad (2)$$

Intuitively, $d_{\min}$ is the distance between p'_j and the closest point on any of the lines in the reference scan. Distance measurement noise of the scanners can be approximately modeled as Gaussian, but there are additionally extreme errors due to occlusion effects or multi-reflections, which prohibit using the sum of distance squares as an error function. Thus, in order to suppress erroneous point-to-line correspondences, we use robust least squares (Triggs et al., 2000) and compute Q as follows:

$$Q(\Delta u, \Delta v, \Delta\varphi) = \sum_j \exp\left(-\frac{d_{\min}(\vec{p}'_j(\Delta u, \Delta v, \Delta\varphi))^2}{2 \cdot \sigma_s^2}\right) \qquad (3)$$

where σ_s^2 is the variance of the laser distance measurement, specified by the manufacturer. This equation takes into account the distribution of the distance measurement values, while suppressing deviations beyond this distribution as outliers. It has least squares-like behavior in the near range, but does not take into account points that are far away from any line. Thus, only the 'good' scan points contribute to this quality function, and we do not have to eliminate outliers prior to the matching. The block diagram of this quality computation is shown in Fig. 4.

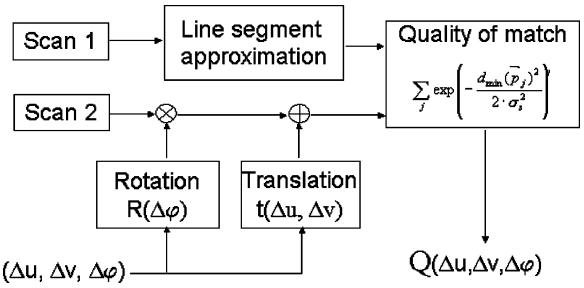

Figure 4. Block diagram of quality computation.

The parameters $(\Delta u, \Delta v, \Delta\varphi)$ for the best match between a scan pair are found by optimizing Q. Steepest decent search methods have the advantage of finding the minimum fast, but due to noise and erroneous point-to-line assignments, they can become trapped in local minima if not started from a "good" initial point. Therefore, we use a combined method of sampling the parameter space and discrete steepest decent, where we first sample the parameter space in coarse steps and then refine the search around the minimum by steepest decent. Hence, we obtain a relative pose estimate $(\Delta u, \Delta v, \Delta\varphi)$ between two scans, to which we refer in the following as a "step". For a series of successive scans indexed by k, we can compute a series of steps $(\Delta u_k, \Delta v_k, \Delta\varphi_k)$, denoting the relative pose between scan k and scan $k+1$.

To reconstruct the driven 2D path, we start with an initial position (x_0, y_0, θ_0), perform a scan match for each step k, and concatenate the steps $(\Delta u_k, \Delta v_k, \Delta\varphi_k)$ to form a path. For non-flat areas, this 2D computation can result in an apparent source of error in length: The scan-to-scan matching estimates for the 3-DOF relative motion, i.e. 2D translation and rotation, are given in the scanner's local scanning plane. If the vehicle is on a slope, this local coordinate system is tilted at an angle towards the global (x, y) plane, and hence the translation should strictly speaking be corrected with the cosine of the pitch angle. Fortunately, the stretching effect is small, and the relative length error is given by:

$$\frac{\Delta l_{\text{err}}}{l} = 1 - \cos(pitch) \approx 1 - (1 - pitch^2) = pitch^2 \qquad (4)$$

While for an impressive 10% slope, the relative error is 0.5%, for a moderate 2% slope, it only amounts to 0.06%. Thus, it turns out that this error is easily within the correction capability of our global localization introduced in the next section. Hence, we utilize the relative estimates from the scan-to-scan matching as if they were given parallel to the ground plane, and concatenate the steps $(\Delta u_k, \Delta v_k, \Delta\varphi_k)$, so that the next global pose $(x_{k+1}, y_{k+1}, \theta_{k+1})$ of the path is computed iteratively as follows:

$$\begin{aligned} x_{k+1} &= x_k + \Delta u_k \cdot \cos(\theta_k) - \Delta v_k \cdot \sin(\theta_k) \\ y_{k+1} &= y_k + \Delta u_k \cdot \sin(\theta_k) + \Delta v_k \cdot \cos(\theta_k) \\ \theta_{k+1} &= \theta_k + \Delta\varphi_k \end{aligned} \qquad (5)$$

As the frequency of horizontal scans is 75 Hz, assuming the maximum city driving speed of 25 miles per

hour, the maximum relative displacement for successive scans is $\Delta u_{\max} = 25$ mph/75 Hz $= 14.8$ cm. In order to reduce the number of scan matches necessary for the path reconstruction, it is advantageous to only use scans with a certain minimum displacement from each other; in particular, this increases computational efficiency and accuracy for times when the vehicle is stationary due to traffic conditions. To obtain steps of approximately equal size, we adaptively subsample the horizontal scans such that we only match pairs that are between 80 and 150 centimeters apart from each other: based on the result of the previous scan match, we predict the next scan likely to have a relative displacement in the desired range, and match this scan with its predecessor; if the result is not in the desired range, we adjust the prediction and redo the computation. Figure 5 shows the resulting path together with the horizontal scan points drawn for each position. As seen, the scan points align well with each other to form footprints of the building facades, confirming the high accuracy of the estimates with respect to each other.

Despite this local accuracy, even small errors in the relative estimates, especially inaccurate angles, accumulate to significant global pose errors over long driving periods. As an example, Fig. 6 shows a 6-km path overlaid on top of a digital roadmap. As seen, the path follows the road map well for the first few hundred feet; nonetheless, the further we drive, the more the reconstructed path deviates from the roads on the map, and at the end, it is entirely off the road. We refer to the globally erroneous path from concatenating the scan-to-scan matching results as initial path; in the next two sections, we devise methods to refine the initial path by using global information.

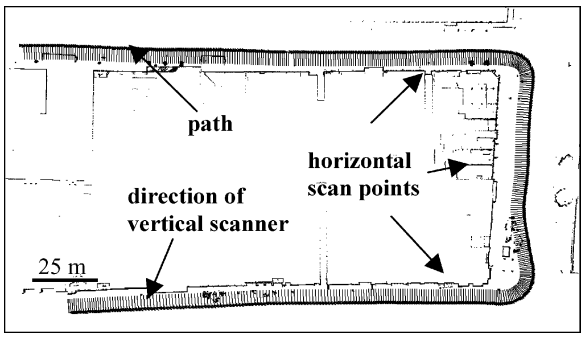

Figure 5. Reconstructed initial path from scan-to-scan matching, and superimposed horizontal scan points for each position.

Figure 6. Initial path obtained by adding relative steps, overlaid on top of a digital road map.

4. Global Maps from Aerial Images or DSM

In this section, we derive a global edge map either from an aerial photo or from a DSM, and define a congruence coefficient as measure for the match between ground-based scans and global edge map. The basic idea behind our position correction is that objects seen from the road-based data acquisition must in principle also be visible in the aerial photos or the DSM. Making the assumption that the position of building facades and building footprints are identical or at least sufficiently similar, one can expect that the shapes of the horizontal laser scans match edges in the aerial image or the DSM. Essentially, we use the airborne edge maps as an occupancy grid for global localization, as it has been done for mobile robots with floor occupancy grids in indoor environments (Konolige and Chou, 1999; Thrun, 2001).

4.1. Edge Maps from Aerial Photos

While perspective corrected photos with a 1-meter resolution are readily available from USGS, we choose to use a higher contrast aerial photograph obtained by Vexcel Corporation, CO, with 1-foot resolution. These aerial photos are not ortho-photos, but by superimposing a metric digital roadmap as shown in Fig. 7(a), we have verified that we can ignore the effect of perspective distortion in the ground plane for the area in the dashed rectangle. However, several other errors can occur:

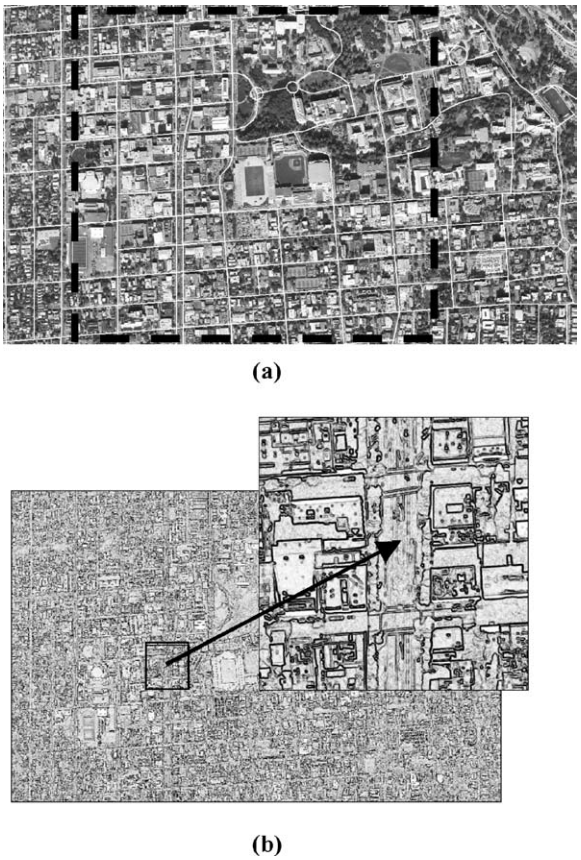

Figure 7. Edge map from aerial photo: (a) aerial photo and (b) edge map.

(1) Only the rooftops are clearly visible in the image, not the footprints. In comparison to the footprints, the rooftops can be shifted by several pixels, depending on the height of the buildings and the perspective.
(2) The photos and the scans were not taken at the same time, so the content can potentially be different. In particular dynamic objects such as cars or buses can cause mismatches.
(3) Visible in the image are not only the building edges, but also many non-3D edges such as road stripes or crosswalk borders. Especially problematic are shadows, because they result in very strong edges.

Despite these potential problems, we assume that in most cases perspective distortion is negligible, and that on average the actual 3D edges, rather than shadows, are dominating. This is because the percentage of tall buildings and shadows in the region we are processing is sufficiently small. In addition, the algorithms introduced in the next section are designed to be rather robust to occasional erroneous edges.

Applying a Sobel edge detector to the aerial image, we obtain a new image where the intensity of a pixel is a function of the strength of an edge. Figure 7(b) shows the resulting edge map and a detailed view.

4.2. Edge Map from DSM

An alternative source for a global edge map is a DSM, which is an array of altitude points uniformly sampled over a 2D area, and as such can be regarded as an airborne height map or a depth image. A DSM can be obtained by manual or automated matching of stereo images, by Synthetic Aperture RADAR, or from airborne laser scans. In our case, a DSM with a one-meter resolution was created using airborne laser scans acquired by Airborne 1 Corp., CA. Implicitly, this DSM defines a gray image where each pixel represents a one by one square meter area on the ground, with a gray value proportional to the altitude at this location.

Using a DSM as a source of a global edge map has several advantages over aerial images: First, it contains two different types of information usable for localization of the data acquisition vehicle, namely (a) the location of building facades as height discontinuities, and (b) the terrain shape and hence the altitude z of the streets the vehicle is driven on. Second, all intensity differences in the DSM are actual 3D discontinuities, whereas for aerial photos, high intensity differences for shadows of buildings or trees result in false edges. Third, there is no perspective shift of building tops to be taken care of; the DSM is virtually a perfect orthoimage. Hence, the major remaining source of error is the potential difference between the roof of a building and its footprint, if the building has an overhanging roof.

While it is possible to apply the Sobel edge detector to the gray level representation of the DSM in order to detect edges, it is advantageous to exploit the altitude information in order to minimize the edge thickness. In the DSM, it is simple to distinguish building tops from ground on the two sides of a discontinuity. Hence, 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. In this manner, only the outermost pixels of taller objects such as building tops are marked, but not the adjacent ground pixels, hence resulting in sharper edges. Furthermore,

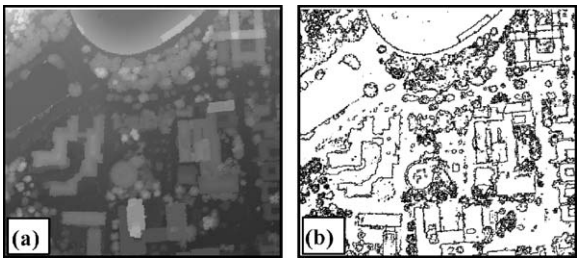

Figure 8. Edge map obtained from DSM with a discontinuity filter. (a) DSM and (b) edge map.

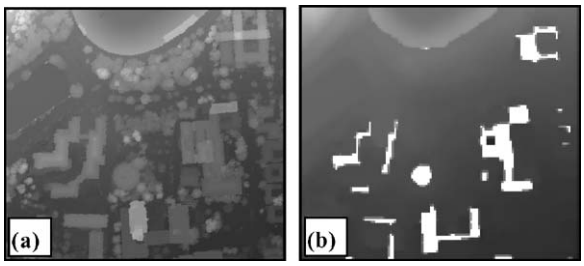

Figure 9. DTM computation: (a) Original DSM, (b) estimated DTM, with blank spots remaining at building locations.

the intensity of an edge is not dependent on the altitude of the building; what matters is whether a discontinuity exceeds the threshold Δz_{edge}. In our case, we choose Δz_{edge} slightly larger than the height at which our sensor unit is mounted, in order to obtain those edges that are likely to be visible in the scans. As shown in Fig. 8, the resulting edge map resembles a global occupancy grid for building walls.

The second type of information in the DSM is the terrain shape. Since our vehicle is always being driven on the road, an estimate of its z coordinate can be obtained from the terrain altitude. Nevertheless, it is not advisable to directly use the z value at a DSM location, since the airborne laser captures overhanging trees and cars on the road at the time of the data acquisition, resulting in z-values of up to several meters above the actual street level for some locations. Thus, we need to create a smooth, dense Digital Terrain Map (DTM) that contains only the altitude of the street level, and we do so in the following manner: Starting with a blank DTM, we first copy all available ground pixels into the DTM. Then, iteratively, we fill the DTM by averaging over the z coordinates of existing ground pixels within a surrounding window, weighing them with an exponential function decreasing with distance. It is not necessary to iterate until all locations are filled; in our application it is sufficient to fill only areas near roads. This is because the DTM altitude at building locations far away from actual ground pixels is not of any interest, since the vehicle has certainly not been driven at these locations. Figure 9(b) shows the resulting DTM, containing ground level estimates of the terrain shape of the roads.

4.3. Congruence Coefficient

Given a 2D pose (x, y, θ) of the truck in the world coordinate system and the corresponding horizontal laser scan, we can transform the local coordinates (u_j, v_j) of the jth scan point into edge map coordinates (x'_j, y'_j) according to

$$\begin{pmatrix} x'_j \\ y'_j \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + R(\theta) \cdot \begin{pmatrix} u_j \\ v_j \end{pmatrix} \tag{6}$$

where $R(\theta)$ is the 2×2 rotation matrix with angle θ. In the edge map, the strength of an intensity discontinuity at the world coordinates (x, y) is expressed as a discretized, integer intensity function $I(x, y)$, with $I(x, y) = 0$ for no discontinuity, and $I(x, y) = I_{\max} = 255$ for the strongest discontinuity. Summing up the intensity values $I(x'_j, y'_j)$ of the edge map pixels corresponding to the N valid scan points of a scan, we define a coefficient $c(x, y, \theta)$ for the congruence between edge image and scan as

$$c(x, y, \theta) = \frac{1}{N} \sum_{j}^{N} \frac{I(x'_j, y'_j)}{I_{\max}}, \tag{7}$$

where $c(x, y, \theta)$ is normalized to the range [0, 1]. Hence, $c(x, y, \theta) = 1$ if all scan points are on an edge of maximum strength, i.e. perfect match, and $c(x, y, \theta) = 0$ if no scan point falls on an edge, i.e. no match at all. Note that in the edge map created from aerial images, $I(x, y)$ is obtained with a Sobel edge detector and thus proportional to the intensity discontinuity in the image. While this is in accordance with the observation that depth discontinuities often result in sharp intensity discontinuities, it is important that no thresholding is applied to the edge image, so as to also utilize depth discontinuities that are less visible in the images. In contrast, for the edge map created from the DSM with our discontinuity filter, all marked edges are true depth discontinuities; $I(x, y)$ is binary and is either $I_{\max}$ for a discontinuity greater than Δz_{edge}, or 0 otherwise.

Figure 10. Edge image with scan points superimposed. The gray pixels denote the edge map and the black pixels denote scan points. (a) pose (x_a, y_a, q_a), with $c(x_a, y_a, q_a) = 0.377$ and (b) pose (x_b, y_b, q_b), with $c(x_b, y_b, q_b) = 0.527$, matching the airborne edge map best.

One can regard the ensemble of laser scan points in the local coordinate system as a second edge image; from this point of view, Eq. (7) essentially computes the correlation between the two edge images as a function of translation and rotation. Therefore, we could also refer to $c(x, y, \theta)$ as a cross correlation coefficient. For the same scan, different global position parameters (x, y, θ) yield different coefficients $c(x, y, \theta)$; the largest coefficient is obtained for the parameter set with the best match between scan and edge map. Figure 10 shows an example for a congruence coefficient for two different pose parameters.

5. Global Correction Based on Monte-Carlo-Localization

In this section, we propose MCL as a robust way to obtain globally correct pose estimates for the acquisition vehicle by using the edge map and the horizontal laser scans. MCL is an implementation of Markov localization, where the environment is assumed to be static and pose is represented by a probability distribution over the parameter space. A motion phase and a perception phase are performed iteratively, both modeled as a stochastic process. It is sufficient to have only a very approximate knowledge about the way the motion and perception affect this distribution. Generally, the motion phase flattens the probability distribution, because additional uncertainty is introduced, whereas the perception phase sharpens the position estimate, because additional observation is used to modify the distribution. As this method can propagate multiple hypotheses, it is capable of recovering from position errors and mismatches.

The crucial point in Markov localization is the implementation, because a reasonable representation for the probability distribution needs to be found. A popular approach in robotics is grid-based Markov localization, where the parameter space is sampled as a probability grid. However, for our downtown area this would lead to more than 10^8 states, and hence large computational complexity, even for resolution as low as one meter by one meter and 2 degrees. Rather, we have chosen to implement MCL, in such a way that the pose probability distribution is represented by a set S of discrete particles P_i, each with an importance factor w_i. These particles are propagated over time using a combination of sequential importance sampling and resampling steps, in short referred to as sampling-importance-resampling. The resampling step statistically multiplies or discards particles at each step according to their importance, concentrating particles in regions of high posterior probability. Hence, the particle density is statistically equivalent to the probability density.

In our particular application, in each step k the set S_k of N particles is transformed into another set S_{k+1} of N particles by applying the following three phases: (a) motion; (b) perception, and (c) importance resampling. Each particle P_i is associated with a specific parameter set $(x^{(i)}, y^{(i)}, \theta^{(i)})$, and the number of particles corresponding to a parameter set (x_k, y_k, θ_k) is proportional to the probability density at (x_k, y_k, θ_k). Therefore, the histogram over $(x^{(i)}, y^{(i)}, \theta^{(i)})$ of the particles approximates the probability distribution of (x_k, y_k, θ_k), and as such, it is this distribution function of the random variable (x, y, θ) that is being propagated from iteration k to $k + 1$ based on the scan-to-scan match in the motion phase, and the scan-to-edge-map match in the perception phase.

Specifically, in the motion phase, we begin with the relative position estimate $(\Delta u_k, \Delta v_k, \Delta\varphi_k)$ obtained from the scan-to-scan matching described in Section 3, and add a white Gaussian random vector to it in order to obtain a new random vector, i.e.

$$(\Delta\tilde{u}_k, \Delta\tilde{v}_k, \Delta\tilde{\varphi}_k) = (\Delta u_k, \Delta v_k, \Delta\varphi_k) + (n(\sigma_u), n(\sigma_v), n(\sigma_\varphi)) \quad (8)$$

where $n(\sigma)$ denotes Gaussian white noise with variance σ^2, and $\sigma_u^2, \sigma_v^2, \sigma_\phi^2$ represent scan-to-scan measurement noise variance, which is approximately known (Früh, 2002). Based on Eq. (6), the parameter set of the ith

particle P_i, is transformed to:

$$\begin{pmatrix} x^{(i)'} \\ y^{(i)'} \end{pmatrix} = \begin{pmatrix} x^{(i)} \\ y^{(i)} \end{pmatrix} + R\left(\theta^{(i)}\right) \cdot \begin{pmatrix} \Delta\tilde{u}_k \\ \Delta\tilde{v}_k \end{pmatrix}$$
$$\theta^{(i)'} = \theta^{(i)} + \Delta\tilde{\varphi}_k, \tag{9}$$

Intuitively, this means that the amount of movement of each particle is drawn from a probability distribution function of the random variable shown in Eq. (8). As a result of this phase, particles that share the same parameter set are "diffused" after the transformation in Eq. (9).

In the perception phase, for each particle with position parameters $(x^{(i)'}, y^{(i)'}, \theta^{(i)'})$, we set a preliminary importance factor w_i^* to the congruence coefficient $c(x^{(i)'}, y^{(i)'}, \theta^{(i)'})$ between laser scans and edge map, according to Eq. (7); we subsequently normalize w_i^* to obtain the importance factor w_i as follows:

$$w_i = \frac{w_i^*}{\sum_{\text{particles}} w_j^*}, \tag{10}$$

Since $c(x^{(i)'}, y^{(i)'}, \theta^{(i)'})$ is a measure of how well the current scan matches the global edge map for a particular vehicle position $(x^{(i)'}, y^{(i)'}, \theta^{(i)'})$, intuitively, the importance factor w_i determines the likelihood that a particular particle P_i is a good estimate for the actual truck position. As such, the importance factor of each particle is used in the resampling phase to compute the set S_{k+1} from set S_k in the following way: A given particle in set S_k is passed along to set S_{k+1} with probability proportional to its importance factor. We refer to the "surviving" particle in set S_{k+1} as a child, and its corresponding originating particle in set S_k as its parent. In this manner, particles with high importance factors are likely to be copied into S_k many times, whereas those particles with low importance factors are likely not to be copied at all. Thus, 'important' particles become parents of many children. This selection process allows removal of 'bad' particles and boosting of 'good' particles, resembling a sort of evolution process. We apply the above three phases at each step k, in order to arrive at a series of sets S_k.

Examples for these sets are shown in Fig. 11: We initialize a set S_0 of N particles by distributing them randomly within an interval $[\pm\Delta x, \pm\Delta y, \pm\Delta\theta] = [\pm 10\,m, \pm\Delta 10\,m, \pm\Delta 10^0]$ around the approximate starting pose in the edge map, as shown in Fig. 11(a). Applying the three phases motion, perception, and resampling for each step k, we arrive at the series of sets S_k, as shown in Fig. 11(b) and (c) for the sets at iterations 30 and 100, respectively. As seen, the blob of particles moves along the path traversed during data acquisition.

It is straightforward to incorporate additional information such as a digital roadmap or eventual GPS readouts during the set computation, in order to constrain parameter space and increase computation speed. Especially for the edge map from the aerial photo, it is reasonable to use the registered digital roadmap in order to restrict positions of the particles to within a few-meter-wide strip around roads. Assigning a zero importance to each "off-road" particle, we can prohibit its selection during the resampling process. Thus, incorrect particles are removed immediately, and much

Figure 11. Sets of MCL particles superimposed over edge map from aerial photo. Particles are shown in black and edges are shown in gray. (a) S_0, (b) S_{30}, (c) S_{100}.

fewer particles are needed to accurately represent the probability distribution.

The blob-like sets S_k represent the probability distribution as to where we believe the vehicle is, and provide a good measure for the range of possible poses. However, to reconstruct a model, we finally have to choose one single pose. In principle, it is possible to extend the consistent pose estimation idea of Lu and Milios (1997b) by the additional constraint that the resulting global pose must be within the range of S_k. While this method could deliver an optimal solution since it considers all subsequent scan-to-scan matches, scan-to-edge map matches, and even the matching between scans when passing a certain street for a second time, it does not scale well to arbitrarily long drives. For our data acquisition described in Section 7, it would result in a 30327-dimensional pose space, and solving a 30327×30327 matrix is impractical even if it is sparse. Therefore, we apply a more heuristic solution, which is linear in path length and delivers sufficiently accurate global poses quickly by adjusting the locally accurate relative poses from scan-to-scan matching according to the statistics of several corresponding sets S_k. More specifically, we use the following strategy to obtain final global pose estimates:

First, we compute an intermediate global pose for each step k as the center of mass of the set S_k; we refer to this intermediate global pose as $(x_{ig}, y_{ig}, \theta_{ig})$. For this computation, we only use those particles, whose descendants have survived M steps later, hence not considering particles revealed as inappropriate by additional data in later steps. This intermediate global pose is globally more accurate than the initial pose obtained via scan matching, referred to as $(x_{\text{init}}, y_{\text{init}}, \theta_{\text{init}})$; however, locally, the $(x_{\text{init}}, y_{\text{init}}, \theta_{\text{init}})$ is more accurate than intermediate global pose $(x_{ig}, y_{ig}, \theta_{ig})$. The most obvious strategy is to use the intermediate global pose to correct for initial pose computed via scan matching. However, we have found this approach not to result in centimeter range accuracy of x and y components, needed in our application. This is primarily due to the gross errors in the angle estimates θ of the initial path. Thus, our approach is to decouple angle correction from x- and y-corrections. Specifically, observe from Eq. (5), that errors in angle at a particular step k only depend on angle errors at previous steps, whereas for x and y, the errors at step k depend on both x, y errors at previous steps, and angle errors at previous steps. This suggests that we should first correct for orientation angles, before correcting x and y estimates. Thus, we compute the orientation angle differences between initial path and intermediate poses, temporally average those differences over several steps, and subtract the averaged differences as corrections to the relative angles of the scan-to-scan matching results. Specifically, if the temporally smoothed version of θ is denoted by θ^L, and $\theta^H = \theta - \theta^L$ denotes the high frequency portion of θ, the new, corrected estimate of θ based on the above description is given by

$$\theta_{\text{corr}} = \theta_{\text{init}} - (\theta_{\text{init}} - \theta_{ig})^L = \theta^H_{\text{init}} + \theta^L_{\text{ig,}} \tag{11}$$

As desired, the above equation implies that the corrected angle contains the local high frequency components of the initial scan matches, and the global low frequency components of the intermediate global pose obtained through MCL. We now re-compute the initial path, using the above corrected angles in Eq. (5), to obtain a new, angle-corrected path, with x, y coordinates denoted by $(x_{\text{new}}, y_{\text{new}})$. Since orientation angle errors have already been corrected, this new path has considerably more accurate x and y values than the initial scan matching path. Thus in applying the same procedure of subtracting temporally averaged differences between the new path and intermediate path (x_{ig}, y_{ig}), we can correct for remaining small position offsets. Specifically, the final corrected x and y values are given by:

$$\begin{aligned} x_{\text{corr}} &= x_{\text{new}} - (x_{\text{new}} - x_{ig})^L \\ &= x^H_{\text{new}} + x^L_{ig} \end{aligned} \tag{12}$$

$$\begin{aligned} y_{\text{corr}} &= y_{\text{new}} - (y_{\text{new}} - y_{ig})^L \\ &= y^H_{\text{new}} + y^L_{ig} \end{aligned} \tag{13}$$

The final result is a 2D path with both local accuracy and correct global pose in all three DOF.

In a flat environment, global pose is completely described by the tuple (x, y, θ). For areas with hills and slopes, it is not possible to estimate beyond those three parameters if we use an edge map from an aerial photo; since it contains only 2D information, the reconstructed path is the projection of the true 3D path onto the 2D ground plane. However, if we use a DSM as a global reference, we can extend the path computation to the 6 DOF necessary in hill areas: Utilizing the additional altitude information the DTM provides, altitude and pitch can be estimated in a simple manner: We can safely assume that the vehicle never leaves the ground while being driven, and thus set the z_k coordinate to the

altitude of the ground level from the DTM at (x_k, y_k) location. Furthermore, the slope is the gradient of the ground level in driving direction; hence, the pitch angle, can be computed as

$$\begin{aligned} pitch_k &= \arctan(slope_k) \\ &= \arctan\left(\frac{z_k - z_{k-1}}{\sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}}\right) \end{aligned} \tag{14}$$

by using the altitude difference and the traveled distance between successive positions. Since the resolution of the airborne scans is only about one meter and the altitude of the ground level is computed using a smoothing process, the estimated pitch does not contain abrupt pitch changes such as those caused by pavement holes and bumps. In spite of this, due to its size and length, the truck is relatively stable lengthwise and as such, the obtained pitch is an acceptable estimate in the long run.

The 6th DOF, namely the roll angle, can similarly be estimated using airborne data, but in this case we have a more accurate alternative: We can assume buildings are generally built vertically, and apply a histogram analysis on the angles between successive vertical scan points. If on average the distribution peak is not centered at 90 degree, the difference between 90 degree and the actual peak angle is an estimate of the roll angle. The result is a 6-DOF path, completely registered with the airborne DSM.

6. 3D Model Generation

Once the pose of the vehicle and thus the laser scanners is known, the generation of a 3D point cloud is straightforward. We calculate the 3D coordinates of the vertical scan points by applying a coordinate transformation from the local to the world coordinate system. The structure of the resulting point cloud is given by scan number and angle, and therefore each vertex has defined neighbors, thus facilitating further processing significantly. In Früh and Zakhor (2002), we propose an entire framework of algorithms to process this point cloud, including foreground identification and removal, erroneous scan point detection, segmentation, hole filling, and facade geometry reconstruction; due to its regular structure, the processed point cloud can easily be transformed into a triangular mesh by connecting adjacent vertices.

Furthermore, camera and laser scanners are synchronized by trigger signals and are mounted in a rigid configuration on the sensor platform. We calibrate the camera before our measurements and determine the transformation between its coordinate system and the laser coordinate system. Therefore, for every arbitrary 3D point, we can compute the corresponding image coordinate and map the texture onto each mesh triangle. The result is textured facade models of the buildings that the acquisition vehicle drove by. Since these facade models have been brought into perfect registration with either aerial photo or DSM, they can eventually be merged with models derived from this same airborne data.

7. Results

The ground-based data was acquired during a 37-minute-drive in Berkeley, California, for which the speed was only limited by the normal traffic conditions during business hours and the speed limit of 25 mph imposed by the city of Berkeley. Starting from Warring Street in the Berkeley Hills, we descended through residential areas, passed through Telegraph Avenue, then further down along Bancroft Way and around the U.C. Berkeley campus. Finally, we drove in clockwise loops around the blocks between Shattuck Avenue and Milvia Street, while always driving two blocks southwards on Shattuck and only one block northwards on Milvia. The driven path was in total 10.2 km long, and while driving, we captured 148,665 vertical and horizontal scans, consisting of 39.74 million 3D scan points along with a total of 7,200 images.

We have applied scan matching and initial path computation to the entire driven path. With a minimum displacement of 80 cm and a maximum displacement of 150 cm for a single step, the adaptive subsampling yields 10109 relative estimates for the entire 10.2 km path. To obtain these 10109 relative steps, 6753 additional intermediate scan matches were computed in the adaptive subsampling and not regarded as a full step, since the distance to the end position of the previous step was too small; thus, we in fact had to perform a total of 16862 scan matches. In previous experiments (Früh, 2002), we have determined that the relative pose obtained via scan matching is typically accurate to 0.03 degree in rotation and 1 centimeter in translation for a normal scan pair with a sufficient amount of 'good' features. However, this accuracy can be less if there are extremely cluttered objects such as

trees present in the scans, for which the obtained scan points are somewhat random. Even worse, if features along one dimension are missing, e.g. in case of a long vertical wall, or if there are no features present at all in a scan, e.g. in a huge parking lot, the relative pose cannot be determined via scan matching. Fortunately, these situations are unlikely, since the scanner's field of view spans a half circle with 160 meters diameter. For our drive, we experienced problems neither in tree areas nor in large parking lots; in the latter case, the scan matching relies entirely on light masts, cable posts, and far-away buildings.

Figure 12 shows the path computed by concatenating the relative steps, superimposed on top of the DSM. The sequence of turns and straight drives is clearly recognizable, and the error in relative pose in the path is locally small. However, it is apparent that the global position becomes increasingly more erroneous as the driving progresses; specifically for the areas traversed multiple times, the individual facades would not match from one pass to another.

To correct global pose via MCL, we first use an edge map from the aerial photo shown in Fig. 7. Since superimposing a digital roadmap revealed that the photo can only be considered a metric map in the rather flat part within the dashed rectangle, we can only correct the 6.7 km long path segment in that area. In each MCL motion phase, for each particle we draw a random pose distortion with a Gaussian distribution and standard deviation $(\sigma_u, \sigma_v, \sigma_\phi) = (3\text{ cm}, 3\text{ cm}, 0.5^0)$ and add it to the relative pose estimate in order to spread out the particles. In the perception phase, we assign each particle the congruence coefficient at its location as importance factor.

We have applied the MCL correction with different number of particles, and found that in areas with clear building structures, it is possible to track the path with 5,000 to 10,000 particles. However, in residential areas with many trees, both false edges in the aerial image and increased inaccuracy in the scan-to-scan matching process contribute to a large pose uncertainty. Especially for one residential area, the number N of particles used had to be chosen quite large in order to represent the spread-out belief distribution sufficiently. In our case $N > 100{,}000$ particles were needed to recover the entire path reliably; this is about two orders of magnitude larger than the "optimal" particles set size of 1000 to 5000 reported in Fox et al. (2000) for a much smaller indoor environment and a hand-made edge map. However, if we use the additional information in the digital roadmap to constrain particle positions to locations within a 25 m-wide strip around roads, the required number of particles drops drastically to only 10,000.

From these particle sets, we compute the global pose based on the approach described in Section 6. Figure 13 shows the superposition of edge map and horizontal laser scans drawn for each of the resulting global path positions. Except for a residential area in the upper left detailed view, where false edges resulting from shadows of trees dominate, the ground-based scan points of the resulting path match the edges in the aerial image quite well for most areas, as seen in the two other detailed views.

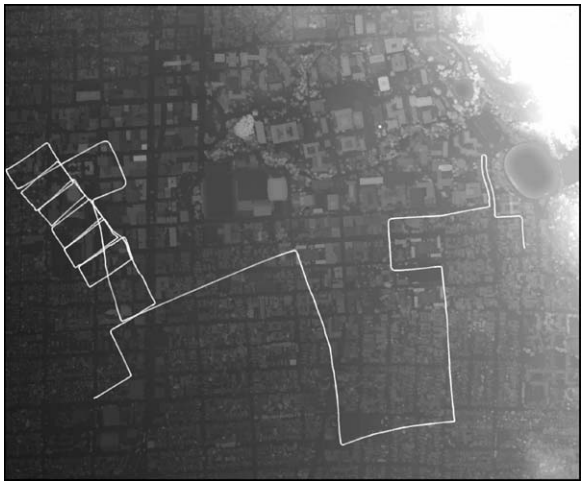

Figure 12. Initial path computed by concatenating relative pose estimates obtained in the scan-to-scan matching process, superimposed on top of the DSM.

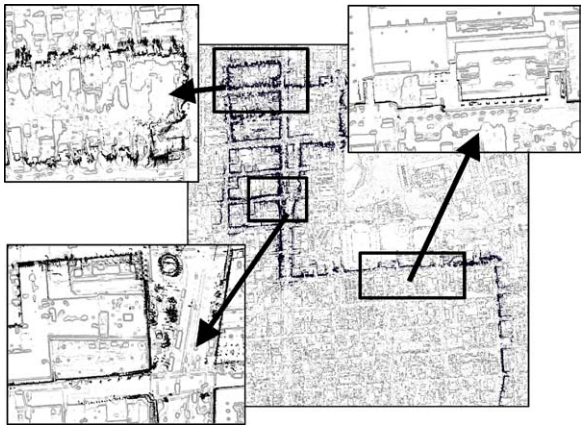

Figure 13. Laser scans projected onto aerial edge image after Monte Carlo localization.

The situation is different for the edge map from the DSM: due to the absence of false edges and perspective distortions, we have found that the vehicle can easily be tracked with as few as 5000 particles. The spread of the particle set is extremely low throughout the entire computation and they are always within the road strip, so that the use of roadmap information is not needed, and provides no benefit. After computing intermediate global pose estimates from the particle sets, we apply the global correction to the initial path: First, we calculate the yaw angle difference between initial path poses and intermediate poses, as shown in Fig. 14(a), and we correct the yaw angles according to the smoothed difference. Then, we recompute the path, and apply the same procedure to the x and y differences, as shown in Fig. 14(b) and obtain the final 2D path. We have

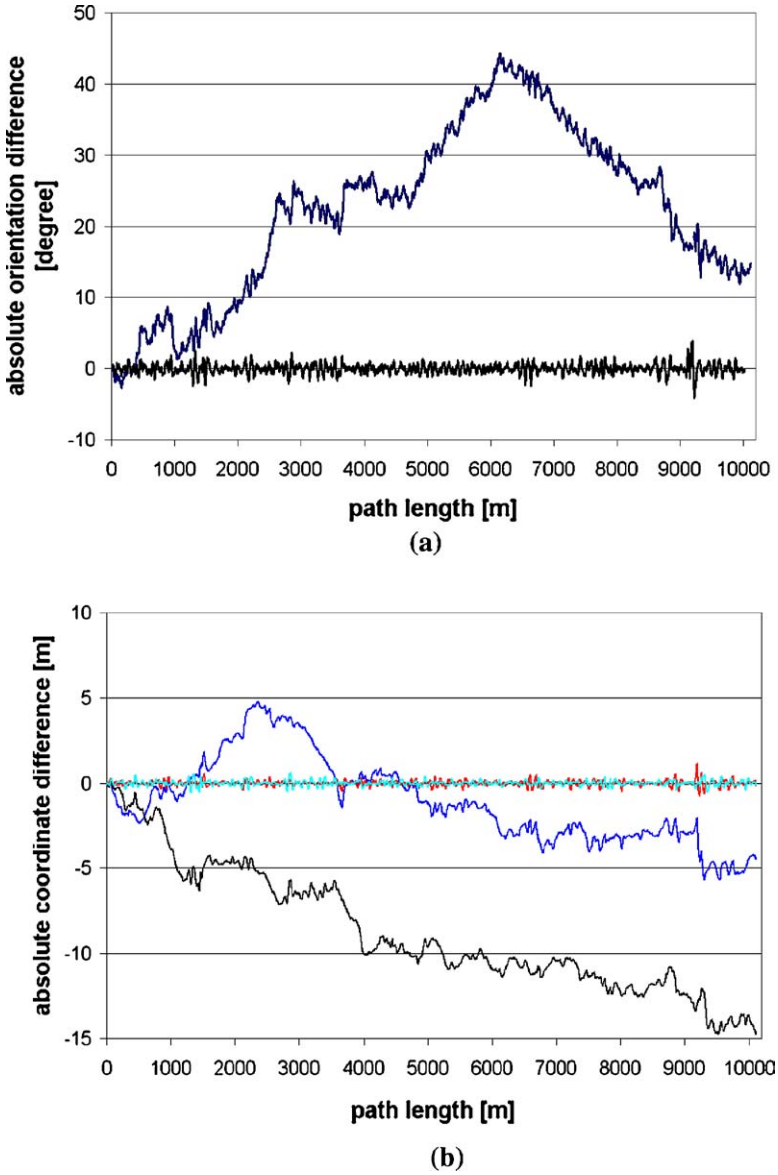

Figure 14. Global correction along the traveled path: (a) yaw angle difference between initial pose and intermediate pose before and after correction, (b) differences of x and y coordinates between angle-corrected pose and intermediate pose, before and after correction. In both diagrams, the differences after corrections are the curves close to the horizontal axis.

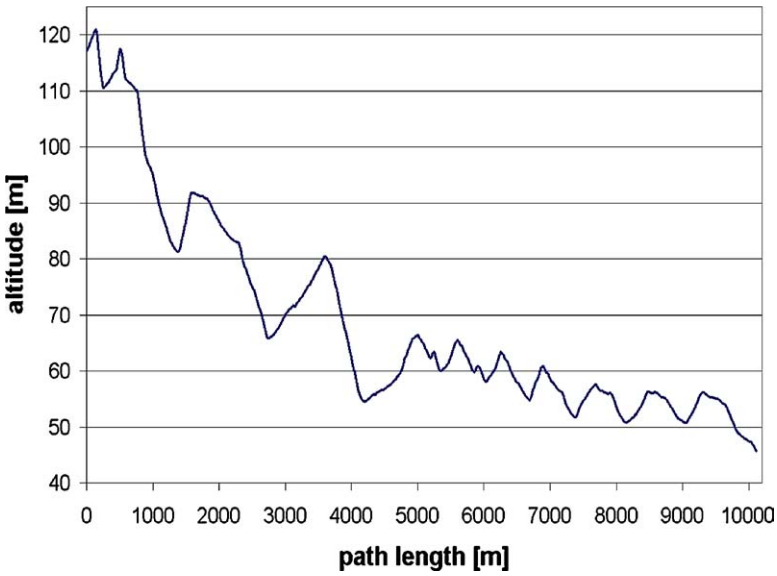

Figure 15. Assigned z coordinates as altitude above sea level.

found that errors in the yaw angles are by far more significant than errors in the x, y estimates: while the initial path is off by several hundred meters, the difference between the yaw-angle corrected path and the intermediate global positions is only a few meters, as seen Fig. 14(b). Finally, to each pose, the corresponding altitude from the DTM is assigned as the z-value, as shown in Fig. 15, and the inverse tangent of the altitude gradient in driving direction, as the pitch angle. The last missing DOF, the roll angle, is computed from the vertical laser scans. The result is a 3D path accurately registered with the airborne DSM.

Figure 16 shows the entire globally corrected 10.2 km path (white), including the hillside segment, superimposed on top of the DSM. As seen, in contrast to the initial path shown in Fig. 12, the corrected one matches the DSM. Accordingly, the ground-based horizontal scan points drawn from the corrected poses match the depth discontinuities in the DSM, as shown in the close-up view of the downtown area in Fig. 17. Furthermore, as seen in Fig. 18 in a close-up view, scan points for the same area align with each other even if they are taken during two different passes. Due to the fact that we have used the same global map for correcting all parts of the path, it is not necessary to explicitly impose the constraint of cross consistency across all scan points; this justifies our computationally simple approach of correcting the relative path estimates with the particle sets S_k.

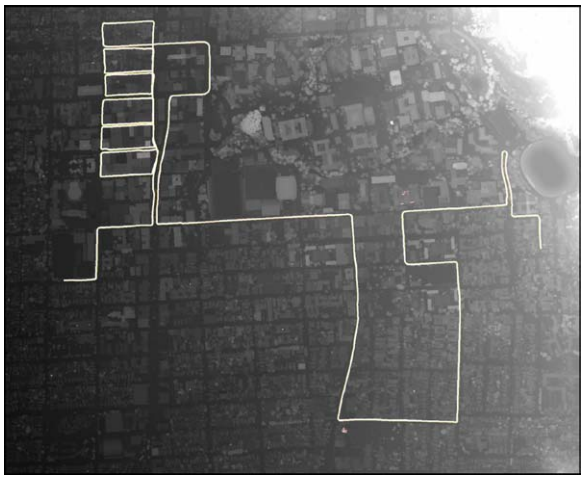

Figure 16. MCL-corrected path superimposed over DSM.

Table 1 shows the processing times for the entire path reconstruction as measured on a 2 GHz Pentium 4 PC. The total processing time is about 4 hours, i.e. 23 minutes per kilometer driven. Since the developed algorithms are linear in path length, our approach scales to arbitrary area size.

Using the accurate pose information, we transform the vertical laser scans into global coordinates and obtain a structured point cloud for a city block as shown in Fig. 19, with a detailed view shown in Fig. 20.

Table 1. Processing times on a 2 GHz Pentium 4 PC.

Processing times to localize the vehicle for the entire 10.2-km/ 37-minutes drive	
Data conversion	32 min
Scan matching and initial path computation	148 min
Monte Carlo Localization (with DSM and 5,000 particles) and global correction	55 min
Total localization time	235 min

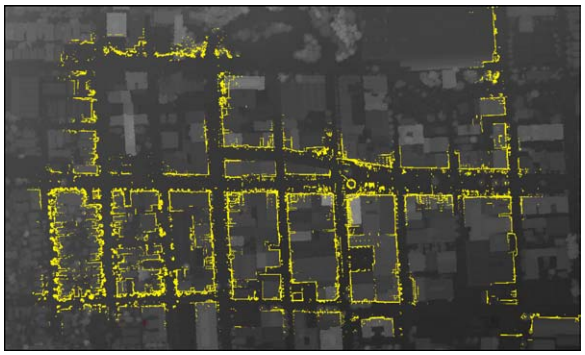

Figure 17. Horizontal scan points for the corrected path superimposed on top of the DSM.

Triangulating the structured point cloud by connecting adjacent vertices, we obtain a surface mesh as shown in Fig. 21. As we have discussed before, our global pose estimation aligns scans from different passes consistently with each other; thus, the resulting facade models align as well. The two facade faces shown in Fig. 22 were taken in two different passes, with several hundred meters driving distance between the two acquisitions. Nevertheless, they align with each other up to a small gap, which is about the spacing between vertical laser scans. This gap occurs because our laser scanner does in general not exactly hit the building at its corner.

Applying the processing algorithms described in Früh and Zakhor (2002), we first divide the path into quasi-linear segments and discard segments that are captured redundantly. Then, for each segment, we remove foreground and fill holes in order to obtain facade models for the downtown Berkeley area as shown in Fig. 23. As seen, the reconstructed building facades are globally registered with respect to the global reference, in this case the aerial photo.

8. Conclusions

We have proposed a method for acquiring ground-based 3D building facade models, which uses acquisition vehicle equipped with two 2D laser scanners. We have demonstrated that scan matching and MCL techniques in conjunction with an aerial photo or a DSM as global map are capable of accurately localizing our acquisition vehicle in complex urban environments. Our method is extremely fast in both acquiring and processing the scan data, and the developed algorithms scale to large environments. However, our localization approach requires an airborne image or a DSM for path correction, which may not be available for some cities

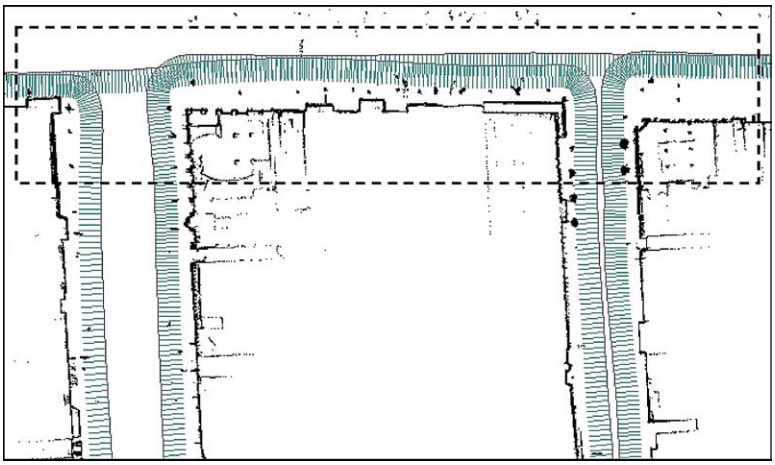

Figure 18. Horizontal scan points of the corrected path. Scan points are drawn in black and the driven path with the direction of the vertical laser scanner is marked in gray. The area inside the dashed rectangle has been passed and scanned multiple times. The scan points inside buildings correspond to interior walls scanned through the windows.

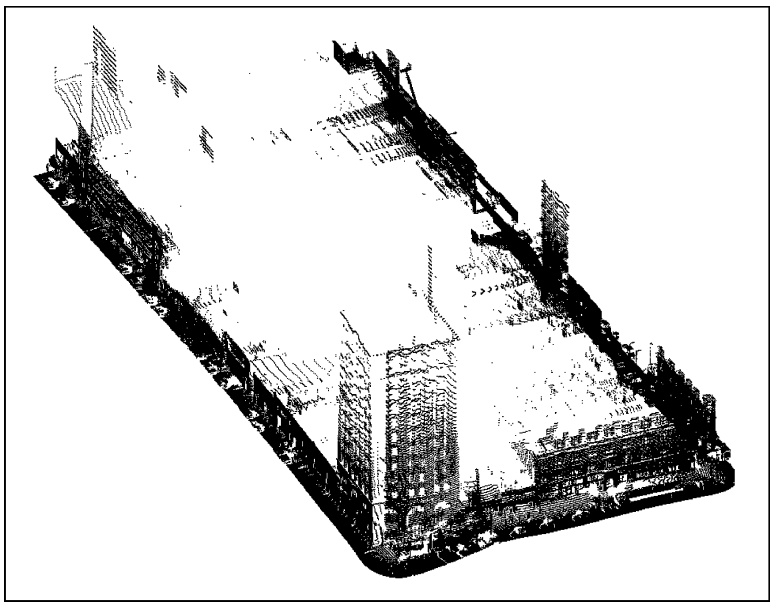

Figure 19. Structured point cloud for a city block.

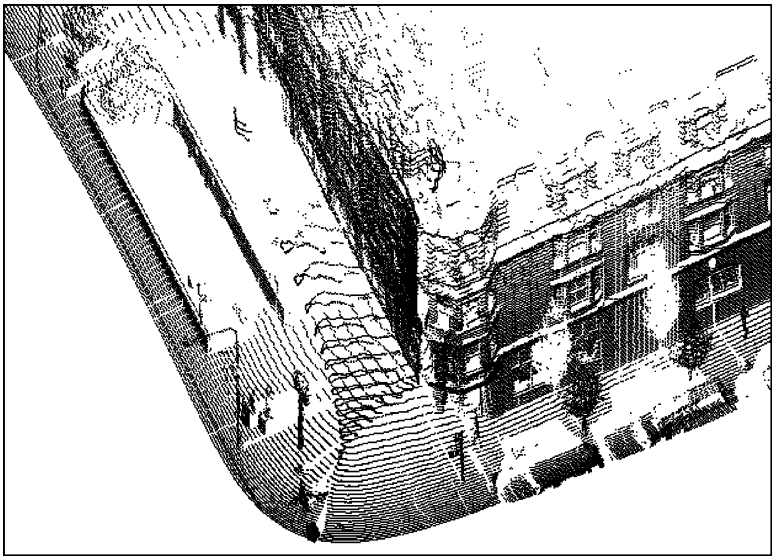

Figure 20. Details of structured point cloud.

or if the vehicle is driven in underground structures such as tunnels or caves. Additionally, in our current implementation, the pose correction is limited to urban areas, since it is dependent on features visible in the scans and the global map. In this sense, it is complementary to GPS, which is more accurate in countryside areas than in urban canyons. If desired, it is straightforward to extend our approach to rural areas by including GPS information during the MCL perception phase. Furthermore, with our truck-based system, we are only capable of driving on roads; hence the facades on the backsides of buildings cannot be captured at all. Finally, the reconstructed raw models are visually not perfect. Foreground objects appear cluttered and visually not pleasing since only their front side is captured, and facades contain large holes due to occlusions or

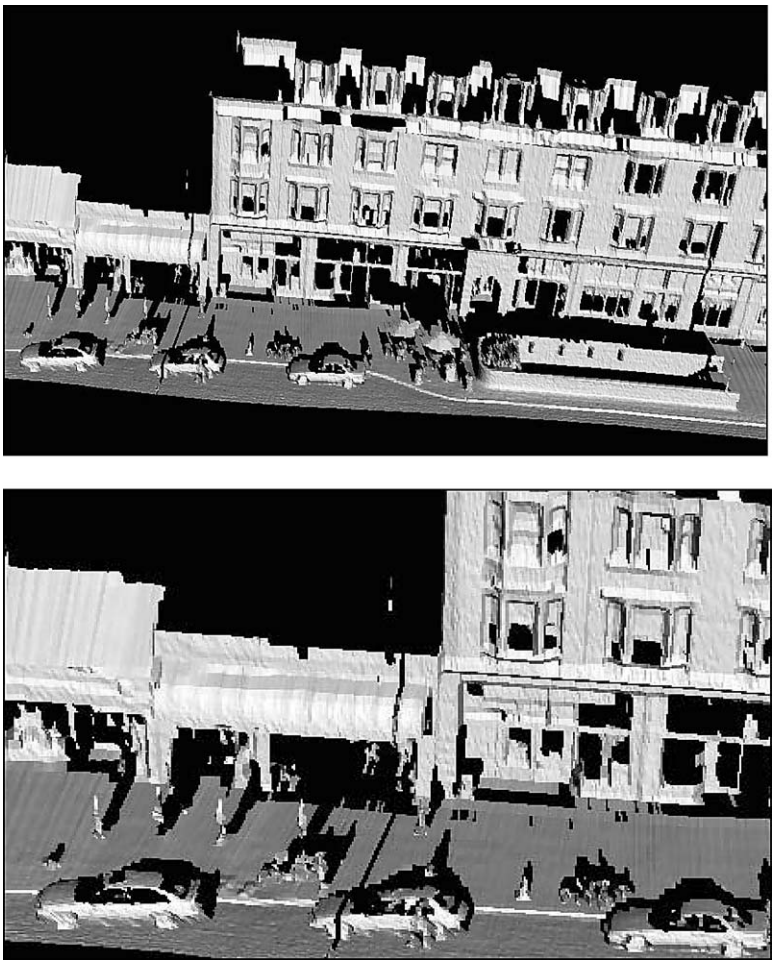

Figure 21. Surface mesh by triangulating the point cloud.

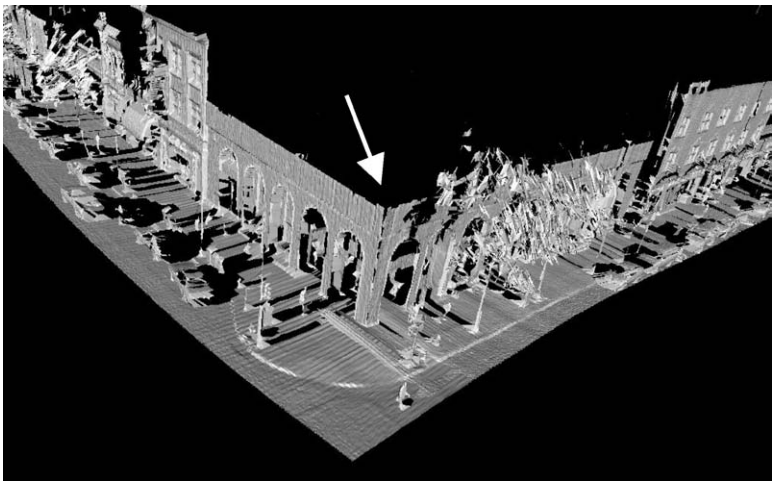

Figure 22. Alignment of two independently acquired facades.

Figure 23. Geometry of facade models overlaid on top of the aerial photo.

reflecting glass surfaces. For downtown areas which can be separated into a facade and a foreground layer, we have proposed processing algorithms for removing cluttered foreground objects and completing occlusion holes in order to obtain visually pleasing facade models (Früh and Zakhor, 2002). However, model reconstruction for more complex building shapes or for residential areas where trees and bushes dominate the scene are still open problems.

References

Antone, M.E. and Teller, S. 2000. Automatic recovery of relative camera rotations for urban scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Hilton Head Island, pp. 282–289.

Besl, P. and McKay, N. 1992. A method for registration of 3-D shapes. *Trans. PAMI*, 14(2).

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

Burgard, W., Fox, D., Hennig, D., and Schmidt, T. 1996. Estimating the absolute position of a mobile robot using position probability grids, AAAI.

Chan, R., Jepson, W., and Friedman, S. 1998. Urban simulation: An innovative tool for interactive planning and consensus building. In *Proceedings of the 1998 American Planning Association National Conference*, Boston, MA, pp. 43–50.

Cox, I.J. 1991. Blanche—An experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7:193–204.

Debevec, P.E., Taylor, C.J., and Malik, J. 1996. Modeling and rendering architecture from photographs. In *Proc. of ACM SIGGRAPH*.

Dellaert, F., Seitz, S., Thorpe, C., and Thrun, S. 2000. Structure from motion without correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Fox, D., Burgard, W., Thrun, S. 1999. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427.

Fox. D., Thrun, S., Dellaert, F., and Burgard, W. 2000. Particle filters for mobile robot localization. In *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon (Eds.), Springer Verlag, New York.

Früh, C. and Zakhor, A. 2001. Fast 3D model generation in urban environments. In *IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems*, Baden-Baden, Germany, pp. 165–170.

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

Früh, C. and Zakhor, A. 2002. Data processing algorithms for generating textured 3D building facade meshes from laser scans and camera images. In *Proc. Int'l Symposium on 3D Processing, Visualization and Transmission 2002*, Padua, Italy, pp. 834–847.

Früh, C. 2002. Automated 3D model generation for urban environments. PhD Thesis, University of Karlruhe, Germany.

Frere, D., Vandekerckhove, J., Moons, T., and Van Gool, L. 1998. Automatic modelling and 3D reconstruction of urban buildings from aerial imagery. In *IEEE International Geoscience and Remote Sensing Symposium Proceedings*, Seattle, pp. 2593–2596.

Gutmann, J.-S. and Konolige, K. 1999. "Incremental mapping of large cyclic environments. In *International Symposium on Computational Intelligence in Robotics and Automation (CIRA'99)*, Monterey.

Gutmann, J.-S. and Schlegel, C. 1996. Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots*.

Hähnel, D., Burgard, W., and Thrun, S. 2001. Learning compact 3D models of indoor and outdoor environments with a mobile robot. *4th European Workshop on Advanced Mobile Robots (EUROBOT'01)*.

Huertas, A., Nevatia, R., and Landgrebe, D. 1999. Use of hyperspectral data with intensity images for automatic building modeling. In *Proc. of the Second International Conference on Information Fusion*, Sunnyvale, vol. 2, no. 2, pp. 680–687.

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

Kawasaki, H., Yatabe, T., Ikeuchi, K., and Sakauchi, M. 1999. Automatic modeling of a 3D city map from real-world video. In *Proceedings ACM Multimedia*, Orlando, USA, pp. 11–18.

Koenig, S. and Simmons, R. 1998. A robot navigation architecture based on partially observable Markov decision process models.

Koch, R., Pollefeys, M., and van Gool, L. 1999. Realistic 3D scene modeling from uncalibrated image sequences. *ICIP'99*, Kobe: Japan, pp. II 500–504.

Konolige, K. and Chou, K. 1999. Markov localization using correlation. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm.

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

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

Lu, F. and Milios, E. 1997. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349.

Maas, H.-G. 2001. 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.

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

Russell, S.J. and Norvig, P. 1995. Articial Intelligence: A modern approach. Chap. 17, *Series in Articial Intelligence*, Prentice Hall.

Seitz, S. and Dyer, C. 1997. Photorealistic scene reconstruction by voxel coloring. In *Proc. CVPR 1997*, pp. 1067–1073.

Simmons, R. and Koenig, S. 1995. Probabilistic robot navigation in partially observable environments. In *Proc. of International Joint Conference on Artificial Intelligence*, Montreal, vol. 2, pp. 1080–1087.

Stamos, I. and Allen, P.E. 2000. 3-D model construction using range and image data. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, Hilton Head Island, pp. 531–536.

Szeliski, R. and Kang, S. 1995. Direct methods for visual scene reconstruction. *IEEE Workshop on Representation of Visual Scenes*, pp. 26–33.

Thrun, S. 2000. Probabilistic algorithms in robotics. AI Magazine, vol. 21, *American Assoc. Artificial Intelligence*, pp. 93–109.

Thrun, S., Burgard, W., and Fox, D. 2000. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proc. of ICRA 2000*, San Francisco, vol. 1, no. 4, pp. 321–328.

Thrun, S., Fox, D., Burgard, W., and Dellaert, F. 2001. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence Journal*.

Thrun, S. 2001. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363.

Triggs, B., McLauchlan, P.F., Hartley, R.I., and Fitzgibbon, A.W. 2000. Bundle adjustment—A modern synthesis. In *Proc. International Workshop on Vision Algorithms*, Corfu, pp. 298–372.

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

Zhao, H. and Shibasaki, R. 1999. A system for reconstructing urban 3D objects using ground-based range and CCD images. In *Proc. of International Workshop on Urban Multi-Media/3D Mapping*, Tokyo.