

3D Model Generation for Cities Using Aerial Photographs and Ground Level Laser Scans

Christian Frueh and Avideh Zakhor¹

*Video and Image Processing Lab
University of California, Berkeley*

In this paper we describe techniques for 3D textured model construction of urban areas using acquisition devices such as intensity cameras, as well as 2D laser scanner. Our experimental set up consists of a truck equipped with one camera and two fast, inexpensive 2D laser scanners, traveling on city streets under normal traffic conditions. The horizontal laser scans are used to determine the approximate component of motion along the movement of the acquisition vehicle. The vertical scanner is used to build 3D models of the facade of the buildings. To improve the accuracy of localization of the truck and hence our resulting 3D models of the city, two different methods are developed and compared: the first method employs a correlation technique and the second method is based on Monte Carlo localization. Both techniques use digital road maps and aerial photographs in conjunction with laser scans. A fairly accurate textured, 3D model of downtown area has been acquired in a matter of few minutes, limited only by traffic conditions during the data acquisition phase.

Keywords: self-localization, scan matching, aerial photos, 3D model generation, urban simulation

I. 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. Commercially available models typically take months to create and usually require significant manual intervention. This process is not only prohibitively expensive, but also is unsuitable in applications where the goal is to monitor changes over time, e.g. detecting damage or possible danger zones after catastrophes such as earthquakes, land slides or hurricanes.

There exist a variety of approaches to creating 3D models of cities. One approach is remote sensing, where satellite or aerial images are used [4, 6] by stereo vision or synthetic aperture radar algorithms. Although these methods can be reasonably fast, the resulting resolution of the models is not high, and without manual intervention, the resulting accuracy is poor. Specifically, they lack the level of detail that is required for realistic virtual walk throughs or drive throughs. In [2] 3D models are created using lines extracted from merged camera images. [3, 10] use laser

scanners mounted on a mobile robot for scanning buildings, but the time required for data acquisition of a entire city is prohibitively large; also, the reliability of autonomous mobile systems in outdoor environments is still relatively poor.

In [5] we propose a method that is capable of rapidly acquiring 3D model and texture data of an entire city at the ground level by using fast 2D laser scanners and digital cameras. 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. The data acquisition system is mounted on a truck moving at a normal speed on public roads, collecting data to be processed offline. Relative position changes can be computed with centimeter accuracy by matching successive laser scans against each other; however, small errors and occasional mismatches can accumulate to a significantly large level in global position estimation of the acquisition vehicle, and hence the 3D reconstructed models, after only a few hundred meters.

On the other hand, there are digital roadmaps and perspective corrected aerial photos available that have poorer resolution, i.e. in meter range, but provide a geometrically correct view over the entire city area. As such, it is conceivable to use them in order to arrive at global position without use of accurate GPS devices. Another advantage of using aerial photos and digital roadmaps over GPS is that the same set of aerial photos can potentially be used to derive approximate 3D models of a city, which can then be merged with the 3D façade model obtained from ground level laser scans.

In this paper, we propose a number of methods to register 3D models built from laser scans with 2D aerial photographs and roadmaps so as to accurately determine the location of our acquisition vehicle in global world coordinates, hence to build globally accurate 3D models. Our approach is to match features seen in both the laser scans and the aerial images, whereby the images can be regarded as a global map onto which the scan points have to be registered. This problem is similar to localization in mobile robotics, where a map of the environment is known, the robot has an approximate estimate of its position usually derived from the odometry, and laser scans are registered with the world model. However, our problem is different from the standard localization problem, because unlike

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

indoor modeling, distances involved in making 3D models for cities are large compared to the range of the laser scans. The outline of this paper is as follows: Sections II and III describe our data acquisition system, and a relative position estimation algorithm for laser scans. In section IV and V we discuss two methods for registration of laser scans with aerial images and digital roadmaps for correcting relative position in order to estimate global position. Section VI demonstrates the final 3D façade model generation.

II. SYSTEM OVERVIEW

As described in [5], the data acquisition system is mounted on a truck and consists of two parts: a sensor module and a processing unit. The processing unit consists of a dual processor PC, large hard disk drives, additional electronics for the power supply and signal shaping; the sensor module consists of two 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 ± 6 centimeters. Both 2D scanners are facing 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 process.

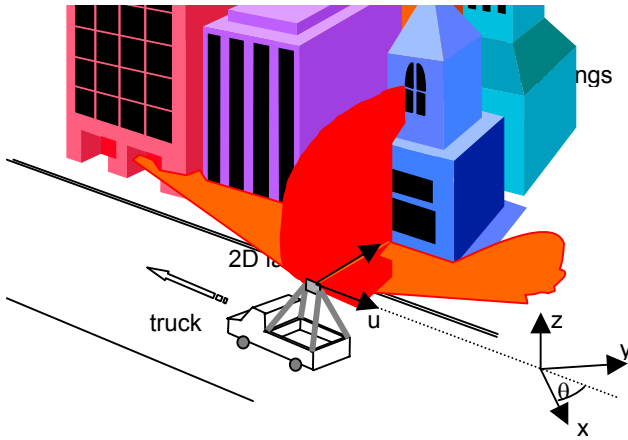


Figure 1: Experimental setup

The vertical scanner detects the shape of the building facades while driving by; the horizontal scanner operates in a plane parallel to the ground and is used for position estimation by scan-to-scan matching. The camera's line of sight is the intersection between the orthogonal scanning planes. Additionally, there is a heading sensor fixed on the sensor plate in order to determine its orientation and a true ground speed sensor. Figure 2 shows a picture of the truck with rack and equipment.

III. RELATIVE POSITION ESTIMATION

First we introduce a Cartesian world coordinate system $[x,y,z]$ where x,y is the ground plane and z points into the sky; and a truck coordinate system $[u,v]$ is implied by the current horizontal laser scan and is therefore parallel to the xy plane as shown in Figure 1. Assuming that the city streets are flat, the position of the truck can be described by the two coordinates x,y and the orientation angle θ of the truck coordinate system.



Figure 2: Truck with acquisition equipment

Rather than estimating the relative movement from odometry, we derive it from scan-to-scan matching [5]. The horizontal 2D scans are captured continuously while driving, and as such, successive scans overlap significantly. Taking one scan as reference, we can approximate it with a series of line segments and match the points of a successive scan to these lines. The displacement Δu , Δv and rotation $\Delta \phi$ between the two scans in the truck coordinate system is estimated by maximizing a quality function $Q = f(\Delta u, \Delta v, \Delta \phi)$, which measures their alignment using a robust least squares method, as described in [5]. Figure 3 shows the line segments of the reference scan in gray, along with the points of the second scan in black, before and after matching.

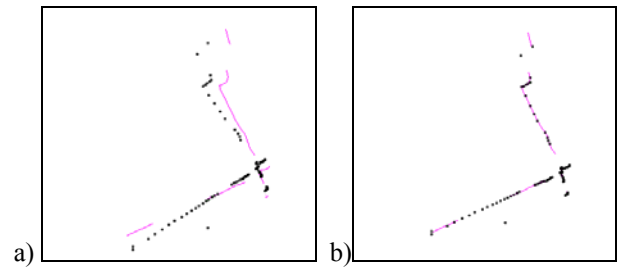


Figure 3: Scan pairs (a) before, (b) after matching

For each pair k of successive scans, we apply our matching algorithm and compute the displacement $\Delta u_k, \Delta v_k$, the rotation $\Delta \phi_k$ in the truck coordinate system, and its quality of match Q_k . From now on, we refer to this as a “step k ” of scan matching. As an initial guess for the actual world coordinates $\{(x_k, y_k, \theta_k)\}$, we can compute a path of

successive positions $\{(x'_k, y'_k, \theta'_k)\}$ in world coordinates as:

$$\begin{pmatrix} x'_{k+1} \\ y'_{k+1} \end{pmatrix} = \begin{pmatrix} x'_k \\ y'_k \end{pmatrix} + R(\theta'_k + \Delta\varphi_k) \cdot \begin{pmatrix} \Delta u_k \\ \Delta v_k \end{pmatrix} \quad (1)$$

$$\theta'_{k+1} = \theta'_k + \Delta\varphi_k,$$

where $R(\theta)$ is the 2×2 rotation matrix with angle θ .

Although the relative position estimates are quite accurate, errors accumulate and the computed positions $\{(x'_k, y'_k, \theta'_k)\}$ can be substantially different from their actual values $\{(x_k, y_k, \theta_k)\}$, and thus an absolute position correction is necessary. In typical mobile robotics application, the same area is often covered repeatedly and therefore there is usually a high degree of overlap among arbitrary scan pairs enabling error correction even after traveling long distances. Unfortunately, this is not the case in our application, as the scan range is small compared to traveled distances, and arbitrary scan pairs do not generally overlap. Therefore, solutions such as Expectation-Maximization map construction cannot be used, and as such we need an estimate for global position instead. In the following sections, we propose a number of methods to match the 3D façade models directly with the aerial images in order to arrive at global position estimation, and hence accurate 3D models.

IV. LOCALIZATION BY MAXIMIZING CROSS CORRELATION

A. Adjustment using digital roadmaps

Digital roadmaps are available for all major cities worldwide; in particular, for the Bay Area in California, roadmaps registered with aerial images can be downloaded from the United States Geographic Survey (USGS)'s web sites. A roadmap can be interpreted as a graph, where every intersection or turn is a node, and the road segment in between is an edge. The important information the digital roadmap can provide is the topology and geometry of the city, and hence the possible driving paths. If for example during a vehicle turn operation an overall estimation error of several degrees occurs, roadmaps can be used to correct the angle. However, the accuracy of this method is limited to the width of the road, which we assume to be unknown. It is especially important that we recover the “high-frequency” component of the vehicle position, e.g. a lane change, because for these cases the resulting 3D model would have incorrect shapes if the driving path is assumed to be a straight line.

In order to find and assign road segments to the traveled path, we make the following assumptions:

1. The starting position is on a road node.
2. The truck can only move along the roads in the map and never off road.
3. Significant changes in driving direction (turns) necessarily occur on road nodes and nowhere else.

Given the path computed from the relative position changes, a line segment approximation A of our traveled path as shown in Figure 4 can be obtained by detecting all turns as major changes of driving directions, and fitting a straight line segment between two turns. Each line l_i has a corresponding driving vector d_i ; the intersection of neighboring lines results in nodes. There can be several nodes in between that are passed without changing driving direction significantly. Therefore we apply a tree search to the digital roadmap to find the node where direction and traveled distance fits best to A .

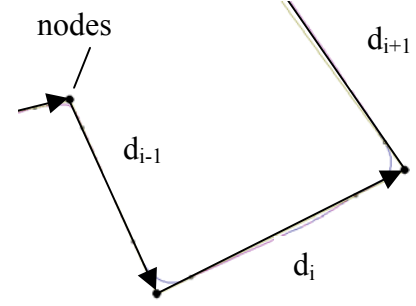


Figure 4: Path and its line segment approximation, A

Our proposed algorithm to track position on the roadmap can be summarized this way:

- 1) Select manually the starting point S of the traveled path on the map.
- 2) Choose the first driving vector d_1 in A and add it to S in order to determine the goal point G on the map:
- 3) Start a tree search in the roadmap that finds all possible road paths that have approximately the same direction as d_1 and no major direction changes between successive road edges.
- 4) Compute the Euclidean distance of each road node N_k passed in 3) to the goal point G and find the one with the shortest distance N_{opt} ; this is the most likely end node and as such a correction angle $\Delta\theta$ and a length correction factor η can be computed.
- 5) Stretch the direction vector d_1 by η .
- 6) Rotate all d_j with $j \geq 1$ by the correction angle.
- 7) Take G as new starting point S .
- 8) Repeat steps 1 through 7 for all line segments l_i .

Using this algorithm we obtain a graph on the road map and therefore a correction for the length and the angle of each line segment. These adjustments have to be applied to the initial position estimates $\{(x'_k, y'_k, \theta'_k)\}$, and it is reasonable to take into account the quality value Q_k computed for each step k during the scan-to-scan match described in section III. We assume that length errors occur mainly during long straight paths, whereas orientation errors are mainly made during turns, and hence, we distribute corrections accordingly, weighing them inversely proportional to Q_k .

Even though the result is an adjusted path estimate $\{(x''_k, y''_k, \theta''_k)\}$ that fits to the roadmap, it is still

ambiguous to within the width of the road, i.e. several meters. This becomes particularly noticeable when the acquisition vehicle travels a long distance in between scanning of the two sides of a given road. Therefore, we develop a further correction step that uses features of the map-like aerial photo to refine the position.

B. Refinement based on matching aerial photos with laser scans

While perspective corrected photos with a 1-meter resolution, registered to digital roadmaps, are readily available from USGS, we choose to use higher contrast aerial photographs obtained by Vexcel Corporation, CO, USA, with a 1-foot resolution. Even though these aerial photos are not ortho-photos, we can safely ignore the effect of perspective distortion, as the percentage of tall buildings in the region we are processing is sufficiently small compared to the entire area. In addition, the algorithm introduced in this section is designed to be insensitive to small errors in perspective displacement.



Figure 5a: Original aerial image



Figure 5b: Edge image

We begin registering aerial photos and the digital roadmaps manually by adjusting offset and rotation, in a matter of a few minutes for the entire downtown Berkeley. 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. Making the assumption that the position of building facades and building footprints are identical or at least sufficiently similar, one can expect that the shape of the horizontal laser scans match edges in the image. Applying a simple 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 5a shows the downtown part of our aerial image, and Figure 5b shows the corresponding edge image with details.

Using our aerial photo several errors can occur:

- 1) Only the rooftops are clearly visible in the image, not the footprints. In comparison to the footprints, the rooftops are shifted by a distance dependent on the height of the buildings and the perspective.
- 2) The photos and the scans were not taken at the same time, so the contents can potentially be different. As such, objects such as cars or buses can cause mismatches.
- 3) Not only are the building edges visible in the image, 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 problems we assume that in most cases perspective is negligible and that on average, the actual 3D edges are dominating.

Given the position (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 j^{th} scan point into world coordinates (u'_j, v'_j) and project it onto the aerial image. Summing up the intensity values of the corresponding pixels, we define a coefficient $c(x, y, \theta)$ for the cross correlation between image and scan as

$$c(x, y, \theta) = \frac{\sum_j E(u'_j, v'_j)}{\sum_j E_{\max}}, \quad (2)$$

where $E(u, v)$ denotes the intensity of the edge image at the world coordinate $[u, v]$ and E_{\max} its maximum possible value. Equation 2 essentially computes the cross correlation between projected laser scan points and the edges in the aerial image.

To find the path $\{(\hat{x}_k, \hat{y}_k, \hat{\theta}_k)\}$ with the maximum cross correlation between photo and scan, we assume that the adjusted path $\{(x_k'', y_k'', \theta_k'')\}$ computed in section IVa is close to the actual positions $\{(x_k, y_k, \theta_k)\}$ and therefore search the region within $(\pm \Delta x_{\max}, \pm \Delta y_{\max}, \pm \Delta \theta_{\max})$ of $\{(x_k'', y_k'', \theta_k'')\}$, where $\Delta x_{\max}, \Delta y_{\max}$ are chosen based on maximum road width. Figure 6 shows an example of resulting position using this technique. The black points in Figure 6 show the laser scan points superimposed on the edge image. The rectangular area $\{\pm \Delta x_{\max}, \pm \Delta y_{\max}\}$, for

which $c(x,y,\theta)$ is computed, is also shown, with the original position estimate $(x_k'', y_k'', \theta_k'')$ of the truck at its center. The scan corresponding to $(x_k'', y_k'', \theta_k'')$ shown in Figure 6a has a coefficient of only $c(x_k'', y_k'', \theta_k'') = 0.377$, whereas the maximum cross correlation, 0.527, occurs at $(\hat{x}_k = x_k - 4.0\text{m}, \hat{y}_k = y_k - 4.5\text{m}, \hat{\theta}_k = \theta_k - 2^\circ)$, denoted by the tip of the arrow shown in Figure 6b. As seen, the scan points for the corrected position in Figure 6b fit the edges significantly more closely than the ones for the uncorrected position in Figure 6a.



Figure 6: Edge image with (a) Scan at initial position with $c(x_k'', y_k'', \theta_k'') = 0.377$; (b) at optimal position matching with aerial image with $c(\hat{x}_k, \hat{y}_k, \hat{\theta}_k) = 0.527$

The $\{(\hat{x}_k, \hat{y}_k, \hat{\theta}_k)\}$ are the positions with the highest correlation of scan and photo. As there might be mismatches or noise, the outliers among correction vectors defined as $(\hat{x}_k, \hat{y}_k, \hat{\theta}_k) - (x_k'', y_k'', \theta_k'')$ are removed and subsequently the correction vectors are smoothed to obtain the final estimates $\{(\tilde{x}_k, \tilde{y}_k, \tilde{\theta}_k)\}$ for global position.

V. MARKOV LOCALIZATION FUSING LASER SCANS AND AERIAL IMAGES

In this section, we propose the use of Markov localization instead of optimizing cross correlation as a more robust way to improve our position estimation from laser scans. Markov localization has the advantage that it represents position estimates not only by one single set of parameters, but instead by a probability distribution over the parameter space. A motion phase and a perception phase are performed iteratively, both modeled by a probability distribution. 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 propagates 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 1meter x 1meter x 2° . Rather, we have chosen to implement the Monte Carlo Localization (MCL) [9], in which the probability distribution is represented by a set S of particles P_i , each with an importance factor w_i . MCL is an iterative process in which for 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.

In our particular application, 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_0, y_0, θ_0) is proportional to the probability density at (x_0, y_0, θ_0) . Therefore, the histogram over (x, y, θ) of the particles approximates the probability distribution of (x, y, θ) , 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-photo match in the perception phase.

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

$$(\Delta \tilde{u}_k, \Delta \tilde{v}_k, \Delta \tilde{\phi}_k) = (\Delta u_k, \Delta v_k, \Delta \phi_k) + (n(\sigma_u), n(\sigma_v), n(\sigma_\phi)) \quad (3)$$

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, assumed to be known [5]. According to equation 1, the parameter set of the i^{th} particle P_i , is transformed:

$$\begin{aligned} \begin{pmatrix} x^{(i)'} \\ y^{(i)'} \end{pmatrix} &= \begin{pmatrix} x^{(i)} \\ y^{(i)} \end{pmatrix} + R(\theta^{(i)} + \Delta \tilde{\phi}_k) \cdot \begin{pmatrix} \Delta \tilde{u}_k \\ \Delta \tilde{v}_k \end{pmatrix} \\ \theta^{(i)'} &= \theta^{(i)} + \Delta \tilde{\phi}_k, \end{aligned} \quad (4)$$

Intuitively, this means that the amount of movement of each particle is drawn from a probability distribution function of the random variable shown in equation 3. As a result of this phase, particles that share the same parameter set are “diffused” after the transformation in equation 4.

During the perception phase, for each particle with new position parameter $(x^{(i)'}, y^{(i)'}, \theta^{(i)'})$, we set a preliminary importance factor w_i^* to the correlation coefficient $c(x^{(i)'}, y^{(i)'}, \theta^{(i)'})$ between laser scans and aerial photos, as described in equation 2; we subsequently normalize w_i^* to obtain the importance factor w_i as follows:

$$w_i = \frac{w_i^*}{\sum_{\text{particles}} w_j^*}, \quad (8)$$

Since $c(x^{(i)}, y^{(i)}, \theta^{(i)})$ is a measure of how well the current scan matches to 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 selection 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 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. The selection phase is also referred to as importance sampling,

We apply the above three phases at each step k , in order to arrive at a series of sets S_k . Each S_k is then used to estimate the final global position at step k . To do so, we first determine the particles in S_k whose descendents have survived M steps later, and hence are in the set S_{k+M} . We then compute the center of mass for these particles in set S_k , and use it as the final global position estimate in step k .

VI. 3D MODEL GENERATION

A. Generation of point cloud and mesh

Once position is estimated using the techniques described in the previous sections, generation of a 3D point cloud is straightforward. We calculate the 3D coordinates of the vertical scanner by applying a coordinate transformation from the local to the world coordinate system. Our point cloud is not unstructured, rather, it has a topology given by scan number, angle, and therefore defined neighbors. Based on this, the point cloud can be triangulated by connecting neighbors if certain constraints such as smoothness in depth are imposed. The resulting mesh might still contain many holes that have to be further processed.

B. Texture Mapping

The camera and the laser scanners are synchronized by trigger signal and are mounted in a solid 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 a textured facade of the buildings that the acquisition vehicle drove by.

VII. RESULTS

We drove the equipment apparatus along a path of 6700 meters in downtown Berkeley under normal traffic conditions during business hours. We started from the intersection of Telegraph Ave. with Blake Street, drove around the campus and then in clockwise loops around the block between Shattuck Ave. and Milvia Street, while always driving two blocks southwards on Shattuck and only one block northwards on Milvia. As our devices are mounted only on the right side of the truck, driving in loops is the only way to acquire data for both sides of the streets. The driving duration was 27 minutes, limited by the driving speed during rush hour. This includes a total of 6.5 minutes during which the truck was stationary due to traffic conditions and hence data acquisition turned off. This resulted in a net acquisition time of only 20.5 minutes.

A. Localization with maximizing correlation

1) Adjustment with roadmaps

We compute the initial path estimation based on the scan-to-scan matching as described in section III and adjust this path with our digital roadmap. Figure 7a shows the initial path superimposed on top of the roadmap. The basic shape of the driving path is clearly visible, but the absolute position is completely incorrect after a few hundred meters, mainly due to angle errors in turns. Applying the tree search in section IVa we compute the traveled roads and mark them in the roadmap as shown in Figure 7b. Then the steps of the initial path of Figure 7a are adjusted, using the geometry of the traveled roads in the roadmap, and the resulting path is shown in Figure 7c. The shape of this path matches to the actual roads, especially at the turns. However, between turns it is sometimes more inaccurate than road width could explain, mainly because our adjustment distributes orientation errors only in turns and length errors only on straight stretches.

2) Correction by cross correlation with aerial photos

Figure 8b shows application of cross correlation based correction to roadmap adjusted paths shown in Figure 8a, as described in section IVb. These paths are superimposed on edge images of the aerial photographs. Figure 8c shows the superposition of the computed roadmap in Fig. 8b and the aerial photo.

Notice that the computed path in Fig. 8a is situated outside the edge boundaries of the actual road, and is visibly incorrect. Furthermore, the corresponding laser scans do not match the building edges, confirming global position inaccuracy in Fig. 8a. These problems are clearly absent in Figure 8b.

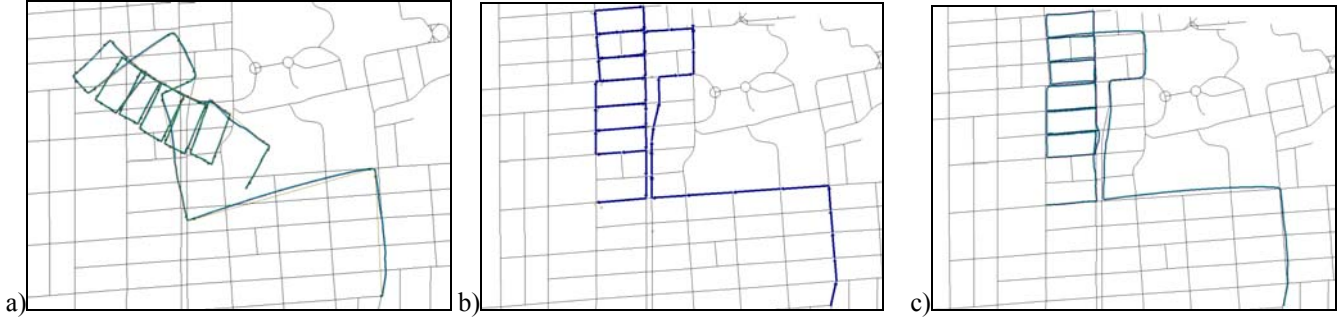


Figure 7: Adjustment with roadmap: (a) original path using relative movements, (b) roads used, (c) path after adjustment

We find that the correction works well in areas with building structures, but not very reliably in areas without distinctive features, e.g. suburban houses hidden among trees. For those areas the algorithm detects that the matching results are not reliable enough and leaves the original uncorrected path unchanged. However, if distinct features are absent in large portions of the path, the accumulated errors can exceed the correlation search range; in this case, once the algorithm is significantly far from the correct path, it may not recover. Therefore the selection of heuristic parameters such as correlation range and correction vector weight, turns out to be crucial for the success of the method.

particles are needed to represent the probability distribution near the roads.

We uniformly distribute a set S_0 of 10000 particles within an interval $[\pm\Delta x, \pm\Delta y, \pm\Delta\theta] = [\pm 10\text{m}, \pm 10\text{m}, \pm 10^\circ]$ around the actual starting position. Figures 9a, 9b, and 9c show the particle set S_k at iterations 0, 30, and 100 respectively. The blob of particles moves correctly along the entire path. From each set S_k we compute one single position estimate $(\hat{x}_k, \hat{y}_k, \hat{\theta}_k)$ as described in section V and use the correction vector averaging described in section IVB to obtain the final path.

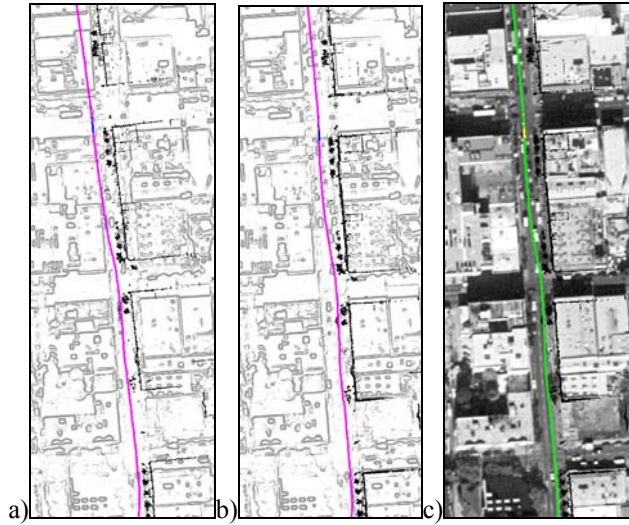


Figure 8: Path and laser scans superimposed on edge images; (a) original path; (b) correlation corrected (c) corrected and superimposed over original image

B. Monte Carlo localization

As we have a digital roadmap registered with aerial photo available, we restrict positions of the particles to within a 25 meters wide strip around roads. Though this is not necessary for obtaining the correct path, it decreases computational time significantly, because incorrect particles can be removed immediately and therefore much fewer



Figure 9: Sets of particles over edge images (a) S_0 ; (b) S_{30} ; (c) S_{100}



Figure 10: Laser scans projected onto aerial edge image after Monte Carlo localization

Figure 10 shows the path and its corresponding laser scan points in black, superimposed with the edge image. It can be seen that these scan points match with edges of the aerial image in most cases, as shown in the lower left detail view. The only area where the matching result did not improve position accuracy is a suburban area, shown in the upper right detail view, where the algorithm tries to match the edges corresponding to shadows of a tree to a tree scan. Despite this temporary loss of accuracy in position estimation, the algorithm recovers as soon as distinct features become apparent.

C. 3D Model Generation

Figure 11 shows a raw façade mesh for the entire path, resulting in a large data size of more than one million triangles. Figure 12 shows the detailed view of a few buildings on Shattuck Ave. The mesh can be further processed, e.g. by filling in windows that appear as holes due to their reflectance, or removing non connected triangles which typically occur when the laser beam hits objects such as cables, light masts etc.

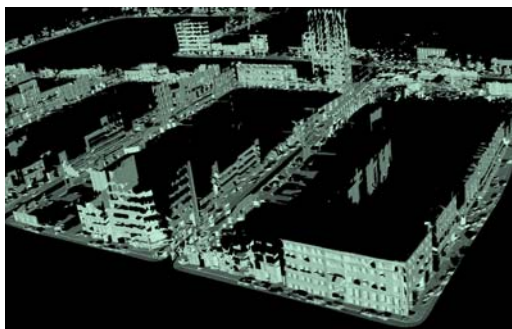


Figure 11: Facade mesh for the scanned roads



Figure 12: Mesh of a single block



Figure 13: Automatically textured mesh

The texture mapping shown in Figure 13 is simply done using the fixed transformation between laser scanner coordinates and camera coordinates, without any sophisticated image processing techniques such as vanishing points or line fitting. Despite the fact that the raw mesh has not been significantly postprocessed, the texture mapping is photo realistic, and results in a reasonable approximation of the 3D scene.

VIII. FURTHER WORK

In future, we will derive full 3D global pose estimation by additional use of the video camera mounted on our truck, thus relaxing the flat area constraint. To fill the space behind the facades invisible from street level, we plan to derive coarse 3D models of buildings by using stereo pairs of the aerial photographs used for localization; as such these models and the photos are inherently registered with each other. Using the methods we described in this paper it is possible to register the laser scans with the photos, and hence to fuse the two models easily.

IX. REFERENCES

- [1] Allen, P. K. and Yang, R.: "Registering, Integrating, and Building CAD Models from Range Data", IEEE Int. Conf. on Robotics and Automation, May 18-20, 1998, Leuven, Belgium, pp. 3115-3120.
- [2] Antone, M.E.; Teller, S. "Automatic recovery of relative camera rotations for urban scenes. " Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, 2000, p.282-9
- [3] Stamos, I.; Allen, P.E. "3-D model construction using range and image data." Proceedings IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, 2000, p.531-6
- [4] Frere, D.; Vandekerckhove, J.; Moons, T.; Van Gool, L. : "Automatic modelling and 3D reconstruction of urban buildings from aerial imagery", IEEE International Geoscience and Remote Sensing Symposium Proceedings, Seattle, 1998, p.2593-6
- [5] Frueh, C.; Zakhor, A.: "Fast 3D model generation in urban environments", IEEE Conf. on Multisensor Fusion and Integration for Intelligent Systems, Baden-Baden, Germany, 2001, p. 165-170
- [6] Huertas, A.; Nevatia, R.; Landgrebe, D.: "Use of hyperspectral data with intensity images for automatic building modeling", Proc. of the Second International Conference on Information Fusion, Sunnyvale, 1999, p.680-7 vol.2. 2
- [7] Sequeira, V.; Goncalves, J.G.M.; Ribeiro, M.I.: "3D reconstruction of indoor environments. " Proc.. Int. Conf. on Image Processing, Lausanne, 1996, p.405-8 vol.2. 3
- [8] Simmons, R.; Koenig, S.: "Probabilistic robot navigation in partially observable environments", Proc. of International Joint Conference on Artificial Intelligence, Montreal, 1995. p.1080-7 vol.2
- [9] Thrun, S. "Probabilistic algorithms in robotics", AI Magazine, vol.21, American Assoc. Artificial Intelligence, Winter 2000, p. 93-109
- [10] Thrun, S.; Burgard, W.; Fox, D.: "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping", Proc. of International Conference on Robotics and Automation, San Francisco, 2000, p.321-8, vol. 1. 4