Fast 3D Model Generation in Urban Environments

Christian Früh and Avideh Zakhor[1]

Video and Image Processing Lab
University of California, Berkeley

This paper describes a new approach to fast generation of three dimensional (3D) models in urban environments. Rather than using a single, expensive 3D scanner, the 3D scene is captured by a combination of two fast, inexpensive 2D laser scanners. Additionally, the data acquisition system consists of a digital camera, a velocimeter and a heading sensor, all mounted on top of a truck which is driven in city streets under normal traffic conditions. One of the laser scanners is used for relative position estimation via a scan matching algorithm, while the other measures the shape of building facades for point cloud generation. The accuracy of the position estimates obtained from laser scans is further improved by taking into account the data from other sensors. Our approach is tested in a real city environment, and the resulting path estimates and 3D models are shown.

Keywords: self-localization, scan matching, 3D model generation, point clouds, urban simulation

I. Introduction

There is a growing demand for three-dimensional (3D) models of urban environments for many applications, including 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 creating and usually require significant manual intervention. This process not only results in high costs inhibiting broad use of the models, but also makes it impossible to use them for applications where the goal is to monitor changes over time, such as detecting damage or possible danger zones after catastrophes such as earthquakes, land slides, and hurricanes.

There exists a variety of approaches to creating 3D models of cities. One approach is remote sensing [5, 9], where 3D models are created by stereo vision or synthetic aperture radar algorithms, using satellite or aerial images. Although with these methods geometric models can be acquired reasonably fast, the accuracy of the resulting models is not very high, and the lack of detailed information and texture renders them unsuitable for some applications such as virtual walk-throughs. In [2] a method is described as how to create 3D models using lines extracted from merged camera images. In [6, 13] 3D laser scanners and in [14] 2D laser scanners are mounted on a mobile robot in order to scan buildings, but the acquisition time for an entire city is prohibitively large. Also, the reliability of such autonomous mobile systems in outdoor environments is still relatively poor.

We propose an experimental set up that is capable of rapidly acquiring 3D and texture data of entire streets from the ground level by using two fast 2D laser scanners and digital cameras. The data acquisition system is mounted on a truck moving at normal speeds on public roads. Data is acquired continuously, rather than in a stop-and-go fashion, and is subsequently processed offline. Rather than solving the 3D scan registration problem associated with 3D scanners, we face a position estimation problem, similar to many applications in mobile robotics. In order to construct an accurate 3D model, the position and orientation of the truck and its sensor unit need to be accurately determined. To this end, we use scan matching algorithms and employ additional sensors for consistency checks. Section II gives an overview of the data acquisition system; in Section III and IV, we describe scan matching and path computation algorithms. Section V shows the results, including the generated 3D point clouds.

II. System Overview

The data acquisition system can be divided into two parts: a sensor module and a processing unit. The sensor module shown in Figure 1 consists of two 2D laser scanners mounted with their scanning planes at 90 degrees, a digital camera, and a heading sensor; the processing unit consists of a dual processor PC, large hard disk drives, and additional electronics for power supply and signal shaping.

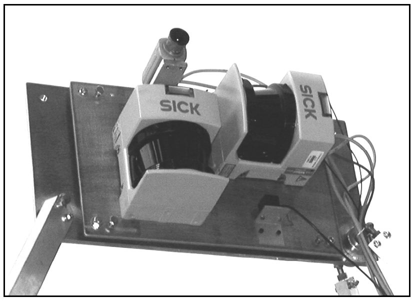

Figure 1: Sensor module

In order to avoid dynamic obstacles such as cars and pedestrians in the direct view, we mount the sensor module

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

on a rack, so that it is at a height of approximately 3.6 meters, whereas the processing unit is mounted onto the truck bed. The scanners have a 180^0 field of view with a resolution of 1^0, a range of 80 meters and an accuracy of ±6 centimeters; the acquisition time for a scan is 6.7 milliseconds. Both 2D scanners are facing the same side of the street; one is mounted vertically with the scanning plane orthogonal to the driving direction, the other one is mounted horizontally with the scanning plane parallel to the ground as shown in Figure 2. During motion, the vertical scanner captures the shape of the complete building facades, whereas the horizontal scanner measures the shape in a plane parallel to the ground, and is used for position estimation as described later.

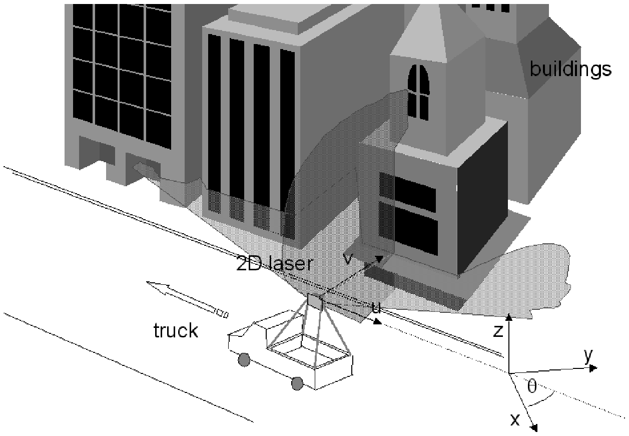

Figure 2: Experimental set up

The camera is mounted towards the scanners, with its line of sight parallel to the intersection between the orthogonal scanning planes. Laser scanners and camera are synchronized by trigger signals, in order to capture at the same position even when the truck is driving fast. Additionally, there is a heading sensor on the sensor plate in order to determine its orientation. A true ground speed sensor (TGSS) is mounted on the back of the car, providing a non-contact speed measurement using Doppler shift. Figure 3 shows a picture of the truck with rack and equipment.

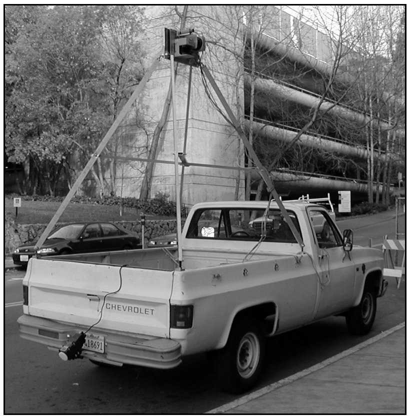

Figure 3: Truck with rack

III. Scan Matching

Laser scanners are most commonly used for applications in mobile robotics, in order to navigate and avoid collisions. If a map of the environment is available, features can be extracted from the scan and their location can be matched against the map in order to estimate the absolute position of the robot. If a map of the environment is not available, both map building and navigation have to be performed iteratively. The preliminary map is used for estimation of the current position and subsequently updated by registration of the new features. In this case, the iterative map building process is critical. As vector estimates of the relative movement are summed up, there is an inevitable accumulation of error over time, unless absolute correction is performed.

In typical mobile robotics application, there is usually a high degree of overlap among arbitrary scan pairs, enabling error correction even after traversing long distances. Unfortunately, this is not the case in our application, as 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. However, since in our application the relative position estimates are accurate enough to recover the shape of the path, it is possible to use GPS or aerial photos for absolute position correction. Indeed, in later phases of our project, we plan to use aerial images to facilitate position correction, and to fuse the models obtained from aerial images with those obtained from 3D laser scans.

In order to obtain relative position change between two captured horizontal scans, they must be registered against each other. Our approach to scan matching is based on line extraction, and is similar to the ones described in [8,11]. Scan points of the reference scan do not necessarily correspond directly to points in the second scan, but can lay in between. Therefore, scan points are not matched directly, rather lines of the reference scan are extracted and matched to either the lines or the points of the second scan. In contrast to many indoor navigation situations, where there are always a sufficient number of lines visible in each scan, the scenery in urban environments is more complex because of many non-planar objects such as trees, masts, cables, and partially reflecting windows. These objects provide additional, sometimes the only information, about the relative position of successive scans, and as such it is desirable to use them for position estimation.

We apply the following algorithm for obtaining a line segment approximation to the reference scan: We connect successive scan points to form a line, only if the difference between their depth values does not exceed a depth dependent threshold. Accordingly, a single scan point can also be approximated by a “degenerate” line segment, i.e. a point. A disadvantage of this method is that straight lines are not computed by a least squares fit over multiple points, but the advantage is that curved objects, such as trees, and small objects, such as masts, cables, are also used for matching. Figure 4 shows the rays of the laser scan in gray and the line segment approximation in black.

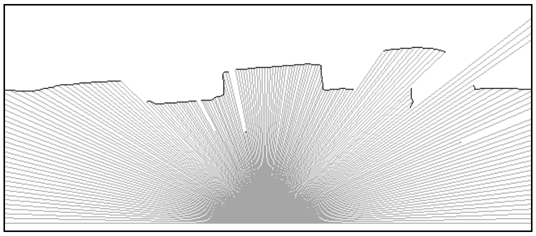

Figure 4: Scan and its line segment approximation

The linear approximation of the reference scan is used as a map to register the second scan. Therefore, we introduce a local coordinate system [u,v] implied by the reference scan, with the sensor module at its center. The u-axis is aligned with the truck's principal axis, while the v-axis is orthogonal to it, with the positive v-axis pointing towards the left side of the truck, as shown in Figure 5. The scanner provides angle and distance for each scan point, enabling us to compute their Cartesian coordinates using simple trigonometric functions. Acquisition time for a single scan is small but since we drive at high speeds, we must compensate for 2D translation and rotation.

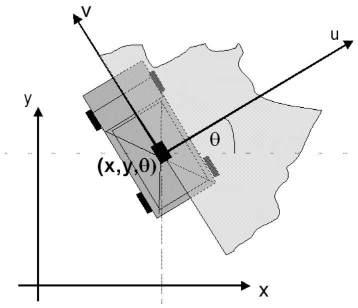

Figure 5: Local and global coordinate system

To match two scans, we maximize a function that computes the quality of alignment Q = f(Δu, Δv, Δφ) for a given displacement Δu, Δv and a rotation Δφ of the scans against each other. Therefore, we perform the following steps: First we compute a set of lines l_i from the reference scan point as described before. Given a translation vector $\vec{t} = (\Delta u, \Delta v)$ and a 2x2 rotation matrix R(Δφ) with rotation angle Δφ, we transform the points p_j of the second scan to the points p'_j according to

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

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:

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

Intuitively, $d_{\min}$ is the distance between p'_j and the closest point on any of the lines in the reference scan. Distance measurement errors of the scanners can be modeled as Gaussian; however, due to outliers resulting from erroneous point-to-line correspondences, resulting from e.g. a perspective change, we use robust least squares [15] rather than least squares to compute Q as follows:

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

where σ_s^2 is the variance of the laser distance measurement, specified by the manufacturer. This formula has least squares-like behavior in the near range but does not take into account points that are far away from any line. The block diagram of this quality computation is shown in Figure 6.

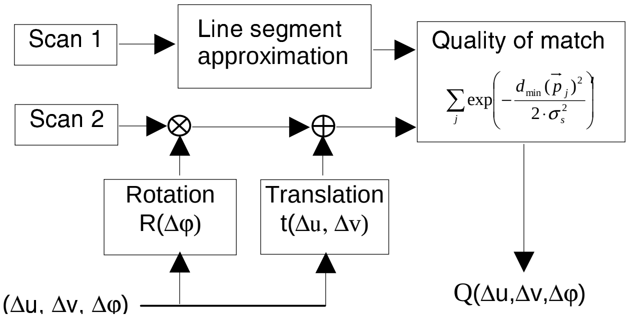
Scan 1
Line segment approximation
Quality of match
$\sum_j \exp\left(-\frac{d_{\min}(\vec{p}_j)^2}{2\cdot\sigma_s^2}\right)$
Scan 2
Rotation R(Δφ)
Translation t(Δu, Δv)
(Δu, Δv, Δφ)
Q(Δu,Δv,Δφ)

Figure 6: Block diagram of quality computation

The parameters (Δu, Δv, Δφ) 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 started from an incorrect initial point. Therefore, a combined method of sampling the parameter space and discrete steepest decent is used, where we sample the parameter space in coarse steps and then refine the search around the minimum with steepest decent. Figure 7 shows the result of such a scan match. The line segments of the reference scan are in gray, along with the points of the second scan in black, before and after matching.

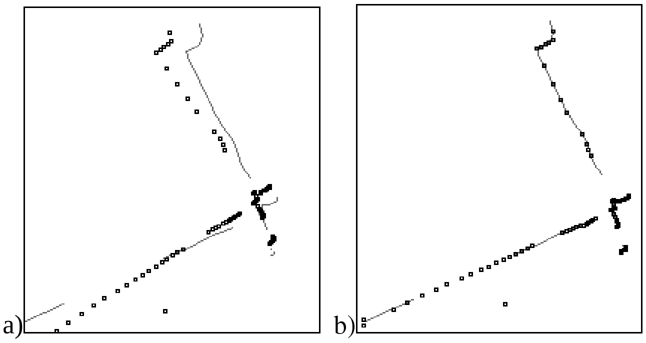

Figure 7: Scan a) before and b) after match

IV. Path computation

As it is impossible to verify position by using the vertical scans without making restrictive assumptions about the environment, a critical part in our approach to creating 3D point clouds is the estimation of the correct position of the sensor module in world coordinates. Since we do not have a sensor providing us with global position estimates at this stage, we have to compute the traversed path by successively adding relative position estimates. These relative estimates, which will be referred to as a step, are derived from the horizontal laser scan matching, and we use other navigation sensors in our system such as the speedometer and the heading sensor for consistency check.

There are different approaches such as Kalman filters to fusing multiple sensors for position estimation in mobile robotics. We have found that the heading sensor and speedometer generally result in much less reliable position estimates than the laser scans. The heading device in particular, is not only noisy, but also has drifts that could not be detected by the filter. In addition, the relative position estimation error of the scan matching does not follow Gaussian white noise statistics; rather, it provides accurate results as long as scan displacements are small; as such, it only occasionally introduces large errors, when there are no significant depth features, e.g. in parking-lots or areas with trees or lawn. Thus, we have opted to implement the following simple but robust rule based navigation module.

The scan matching algorithm determines directly the displacement Δu_i, Δv_i and the rotation $\Delta\varphi_i$ for each step S_i in the local coordinate system; subsequently a consistency check with the other sensors is performed as follows:

1. If there is a significant change in the computed displacement vector, we accept it as valid if both
   a. successive vectors change similarly
   b. the speedometer indicates a speed change

2. If there is a significant change in the orientation angle, we accept the measurement if both
   a. succeeding angles change similarly
   b. the heading sensor indicates an angle change

In all other cases we reject the values and replace them by interpolating between valid preceding and succeeding steps.

We assume that our environment is flat without significant hills, so that we can describe the global pose of the sensor module in a 2D plane by three parameters (x, y, θ) where x,y are the Cartesian world coordinates and θ is the orientation angle of the truck as shown in Figure 5. If speed $V(t)$ and orientation $\theta(t)$ of the truck are known, the motion of the rear axis of the truck can be described as:

$$\begin{aligned} x(t+\Delta t) &= x(t) + V(t)\cdot\Delta t\cdot\cos(\theta(t)) \\ y(t+\Delta t) &= y(t) + V(t)\cdot\Delta t\cdot\sin(\theta(t)) \end{aligned} \quad \textbf{(4)}$$

The sensor module is not mounted above the rear axis, but in the middle of the truck, so that during motion along a curve, it experiences a motion component not only along, but also orthogonal to the truck's principal axis. As an estimate for the relative position change (Δu_i, Δv_i, $\Delta\varphi$) of the sensor module in its local coordinate system [u,v] is obtained for each step S_i, we can compute the global positions (x_i, y_i,θ_i). Therefore, we start with an initial position (x_0, y_0,θ_0), perform for each step S_i a scan match to obtain (Δu_i, Δv_i, $\Delta\varphi$), and apply the coordinate transformation, so that the new position (x_{i+1}, y_{i+1},θ_{i+1}) can be computed in world coordinates as:

$$\begin{aligned} x_{i+1} &= x_i + \Delta u_i\cdot\cos(\theta_i+\Delta\varphi_i) - \Delta v_i\cdot\sin(\theta_i+\Delta\varphi_i) \\ y_{i+1} &= y_i + \Delta u_i\cdot\sin(\theta_i+\Delta\varphi_i) + \Delta v_i\cdot\cos(\theta_i+\Delta\varphi_i) \\ \theta_{i+1} &= \theta_i + \Delta\varphi_i \end{aligned} \quad \textbf{(5)}$$

Since errors in the estimation accumulate with each iteration step of equation 5, it is important to recover the path with as few steps as possible by subsampling the scans by a large factor; this is especially necessary when the truck is stationary due to traffic conditions. On the other hand, it is desirable to use scans that are taken from almost the same position, so that perspective change is negligible, and that overlap between scans is sufficient for accurate matching; this would require the subsampling factor to be small. Therefore, we need a compromise between these conflicting requirements.

As the scanner takes horizontal scans at a frequency of 75 Hz, and assuming the maximum city driving speed to be 25 miles per hour, the maximum relative displacement for successive scans is

$$\Delta u_{max} = 25 \text{ mph}/75 \text{ Hz} = 14.8 \text{ cm.} \quad \textbf{(6)}$$

For typical distances in our measurement scenario, we have found that perspective and missing scan overlap become critical issues for position changes of more than 2 meters. To strike a balance between large and small subsampling factors, we have chosen not to use a fixed subsampling factor, but to adapt it to the driving speed, so that estimated displacements between successive matched scans is between 80 and 100 cm. This not only increases the accuracy of the path, but also improves the computational efficiency.

V. Results

Using the data acquisition system, we have taken measurements in downtown Berkeley, driving around the block Allston Way, Shattuck Avenue, and Center Street, during rush hour with pedestrians and cars scattered all over the place. The total data acquisition time for this block was 203 seconds, limited by the driving speed during rush hour. If the stops at the corner and red traffic lights are subtracted, the net scanning time is 118 seconds. The data was processed as described in sections III and IV and an estimate for the path was computed offline.

Figure 8 shows the computed path using a fixed subsampling factor of 10. The path is represented by a thick black line going around the block; orthogonal to it, the scanning direction of the vertical laser scanner for each computed position is shown. Also shown are the points of the horizontal laser scans for each position, resulting in a footprint of the building facades. The alignment of these scan points is a measure of the computed path accuracy; if the path is correct, the points of a building wall measured from different positions should ideally lay on a sharp line, otherwise they will form a broad, blured strip. The areas where the truck moved slowly or stopped completely can be clearly identified using the drawn vertical scanner directions: positions are computed at fixed time intervals, and therefore the density of estimates is higher during slow motion.

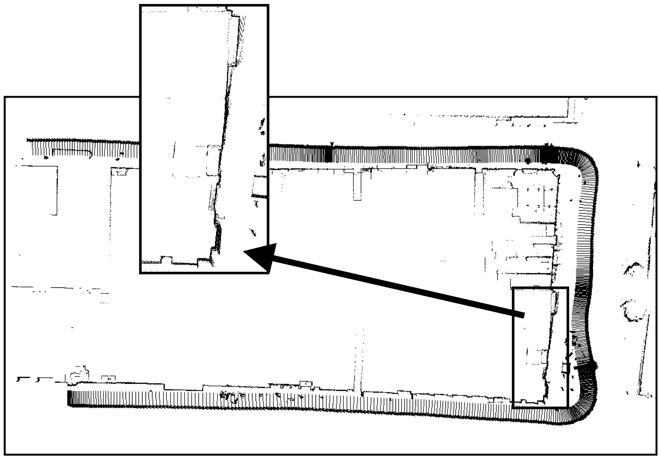

Figure 8: Path computed with fixed subsampling factor of 10

For these parts of the path, the scan point alignment is visibly worse than for the rest, as shown in the detailed view, because the accumulation of the estimation noise leads to position inaccuracy.

Figure 9 shows the same path calculated with adaptive subsampling, given a desired minimum of 80 centimeters per step. As seen in the detailed view, the scan alignment is significantly better, regardless of speed or stopping times during data acquisition. This also results in correct angles between the roads. Note that the upper and lower part of the traveled path are not parallel, because we changed lanes during driving; this lane change occurs in the path as evidenced by the decrease in distance between computed path and building facades from left to right in the upper trajectory of Figure 9. Furthermore, the angle between the two roads and Shattuck Av. on the right side of the Figure 9 is actually not 90^0, also computed correctly by our algorithm.

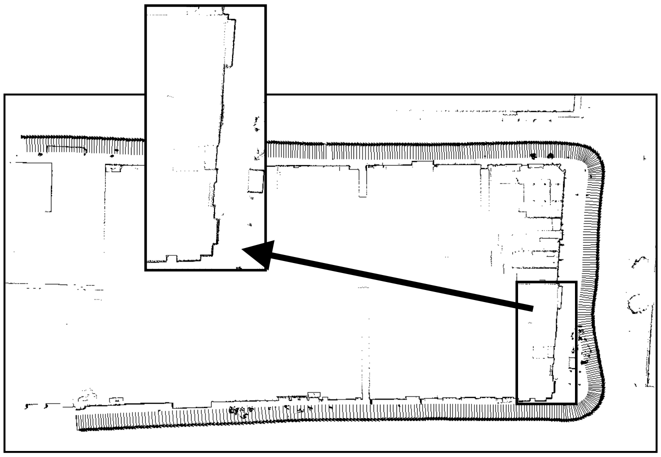

Figure 9: Path computed using adaptive subsampling

Next the vertices of the vertical scans are computed according to the estimated positions and scanning directions. Figure 10 shows the 3D point cloud of the entire block, consisting of 515698 vertices. Details of the point cloud are shown in Figure 11 and Figure 12, where not only the shape of the building facades, but also details such as pedestrians, cars, mail boxes and street lights are visible.

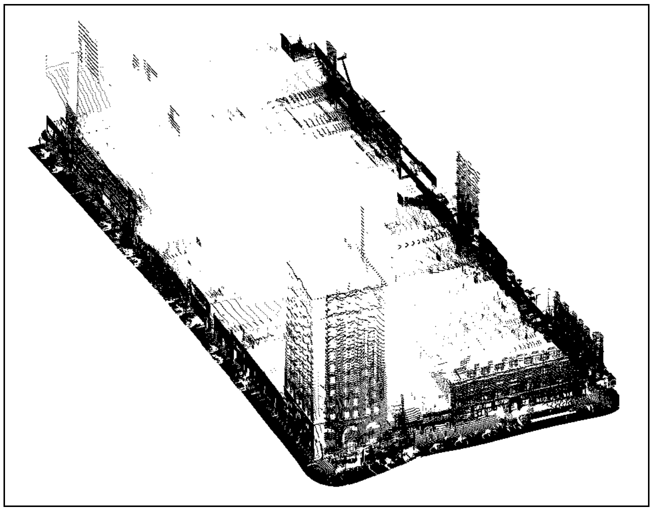

Figure 10: Scan of the entire block

The 3D point cloud can be further processed, by connecting vertices in a mesh, reducing polygons by extracting plane surfaces, and texture mapping in order to obtain a textured façade model shown in Figure 13. As seen, people waiting for the bus are also part of the obtained model, and further processing might detect and remove such objects, if they are not desired.

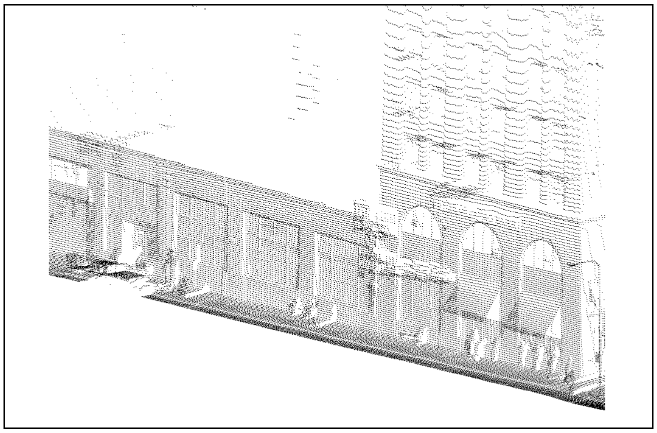

Figure 11: Detailed view (Center street)

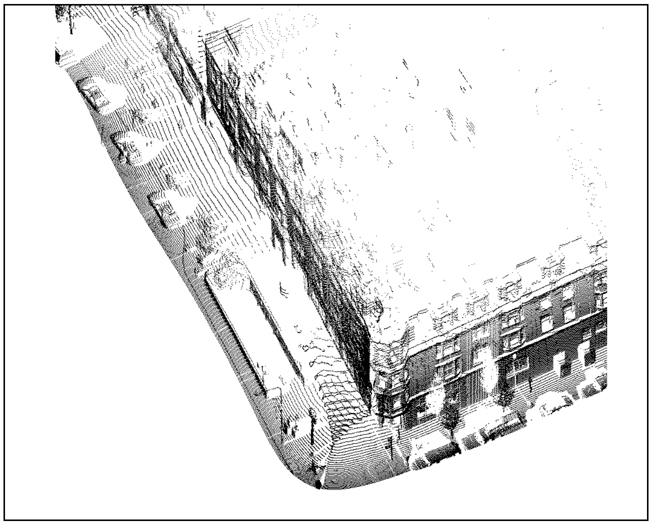

Figure 12: Detailed view (Corner Shattuck, Addison)

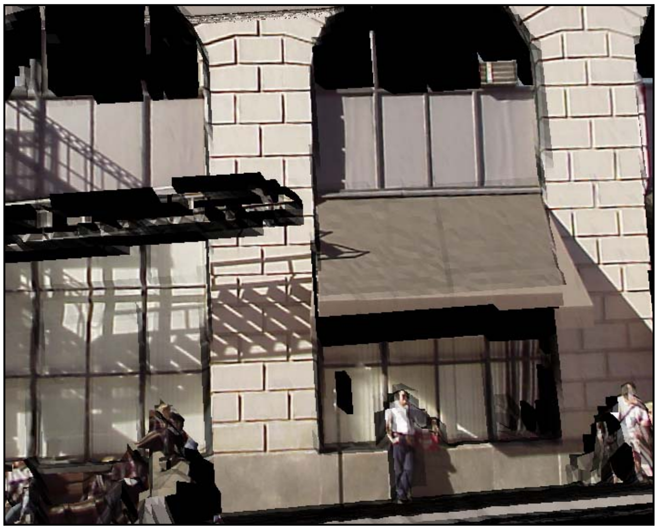

Figure 13: Details of textured facade model

VI. CONCLUSION

We have described a new method to obtain 3D point clouds of urban environments using a system consisting of two 2D laser scanner and additional sensors. The method is implemented in a system mounted on a truck and tested in real city environments. Data acquisition time for an entire street block is significantly lower than that of traditional 3D scanners and only limited by traffic conditions. Further developments include integration of aerial/satellite data to improve absolute position estimation and aid registration of the acquired model into a global map [7].

VII. LITERATURE

[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] Bortwick, S.; Durrant-Whyte, H.: ”Dynamic Localization of Autonomous Guided Vehicles”, Proc. Of Int. Conf. On Multisensor Fusion and Int. Systems, Las Vegas, 1994, p. 92-97

[4] Cox, I.J. “Blanche – An experiment in guidance and navigation of a mobile robot”, IEEE Trans. Robot. Automat., vol 7, pp 193-204, 1991

[5] 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

[6] Frueh, C.; v. Ehr, M; Dillmann, R.: „ Aufbereitung von Laserdaten für ein mobiles autonomes 3D-Meßsystem “, AMS 2000, Karlsruhe, p. 263-270

[7] Frueh, C.; Zakhor, A.: „3D model generation for cities using aerial photographs and ground level laser scans “, submitted to CVPR 2001

[8] Gutmann, J.-S.; Weigel, T.; Nebel, B.: “Fast, accurate, and robust self-localization in polygonal environments”, IROS'99, Kyongju, 1999, p.1412-19

[9] 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

[10] Kam, M.; Zhu, X.; Kalata, P. :”Sensor Fusion for Mobile Robot Navigation”, Proc. Of IEEE, Vol 85, No 1., Jan 1997, p.108 – 119

[11] Lu, F., Milios, E.:”Robot pose estimation in unknown environments by matching 2D range scans”, Journal of. Intel. and Robotic Systems, 18, p.249-275

[12] 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

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

[14] 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

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

[16] Whitaker, R.T.: “Indoor scene reconstruction from sets of noisy range images.“ Second International Conference on 3-D Digital Imaging and Modeling, Ottawa 1999. p.348-57