

# AN INTEGRATED ON-BOARD LASER RANGE SENSING SYSTEM FOR ON-THE-WAY CITY AND ROAD MODELLING

F Goulette, F Nashashibi, I Abuhadrous, S Ammoun, C Laugeau

► **To cite this version:**

F Goulette, F Nashashibi, I Abuhadrous, S Ammoun, C Laugeau. AN INTEGRATED ON-BOARD LASER RANGE SENSING SYSTEM FOR ON-THE-WAY CITY AND ROAD MODELLING. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Copernicus GmbH (Copernicus Publications), 2006, 34 (A). hal-01259660

**HAL Id: hal-01259660**

**<https://hal.archives-ouvertes.fr/hal-01259660>**

Submitted on 21 Jan 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# AN INTEGRATED ON-BOARD LASER RANGE SENSING SYSTEM FOR ON-THE-WAY CITY AND ROAD MODELLING

F. Goulette<sup>\*</sup>, F. Nashashibi, I. Abuhadrous, S. Ammoun, C. Lurgeau

Robotics Lab. (CAOR), Mines Paris,  
60 bd. Saint Michel, 75272 PARIS Cedex 06, France  
goulette@ensmp.fr

**KEY WORDS:** Three-dimensional, Laser Scanning, GPS/INS, Geometric Modelling, On-board processing, Urban, Scene.

## ABSTRACT:

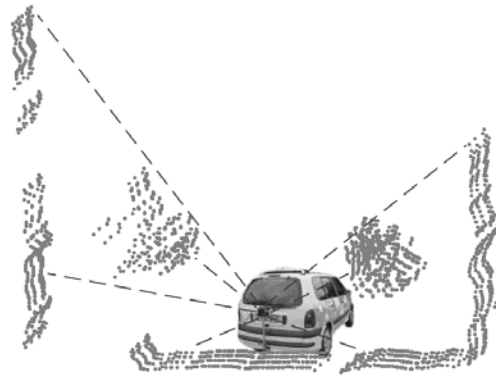
Precise realistic models of outdoor environments such as cities and roads are useful for various applications. However, for a high level of detail, and a large size of environment to be digitized, one has to face the issues of quantity of information to be acquired, processed and stored, and of overall processing time. We present in this paper an integrated on-board laser range sensing system addressing this need: it is designed to perform city and road geometric modelling as it moves. It is based on a laser range sensor mounted on a vehicle whose position is known through GPS-INS localization; it produces raw 3D range data and performs specific modelling for cities and features extraction for roads.

## 1. INTRODUCTION

Models of outdoor environments such as cities and roads are useful for various applications, such as architecture planning, 3D cartography for car and pedestrian navigation, virtual tourism, Virtual Reality, video games, and so on. For realistic modelling, a precise digitizing of 3D geometries and textures is useful, at the level and scale of the desired use (e.g. ground level for car and pedestrian navigation). However, given a desired level of detail and a desired size of environments to be digitized, one has to face the issues of quantity of information to be acquired, processed and stored, and of overall processing time. To obtain 3D geometries of outdoor environments at the ground level, one can use passive methods such as photogrammetry or stereovision to deliver the location of specific points or features in the scene, or active methods using lasers or structured light to deliver 3D points all over a scanned surface. The active methods usually result in more dense data sampling. Several people have explored the use of laser range sensors mounted on vehicles, in order to scan large areas of environments. One possibility is to define several scanning spots, and to make a fusion of the 3D data clouds with adapted frame transformation between the locations, thus requiring post-processing (Allen et al, 2001; Stamos et al., 2000). Another possibility is to use the vehicle itself as a scanning device, its movement defining one of the scanning directions (Früh and Zakhor 2001, 2003, 2004; Zhao and Shibasaki 2001, 2003a, 2003b). In the reported approaches however, the processing of data collected during scanning is done afterwards and is quite heavy, of the order of several hours (Früh and Zakhor 2004; Zhao and Shibasaki 2003a, 2003b), and would not allow on-board processing.

We present in this paper an integrated on-board laser range sensing system, which has been designed specifically for dense 3D digitizing of the geometry of cities and roads, at the vehicle speed. Our system (Figure 1) is an integrated platform, consisting of a car equipped with localization sensors (GPS, IMU) and a laser range sensor. We use normal GPS (and neither DGPS nor RTK) for purpose of low cost solution. The laser range sensor, placed at the rear of the vehicle, performs scanning over a vertical plane perpendicular to the direction of

the car. During the movement of the car, at low or normal speed, the laser is thus scanning the whole environment driven through.



**Figure 1 : LARA-3D,  
on-board laser range sensing prototype**

For the processing of sensor data, we use integrated real-time software designed specifically for our platform, which realizes the functions of generic 3D digitizing (presented in Section 2) and of 3D modelling adapted to cities and road environments (Section 3). The software design has been done to allow on-board real-time processing along the movement of the vehicle; in the text, we assess the precision of the 3D points obtained by the system and its real-time performances.

## 2. ON-BOARD LASER RANGE SENSING

### 2.1 Functional analysis

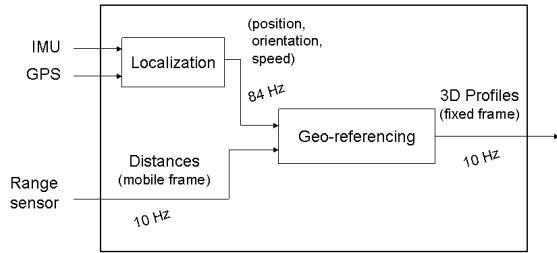
In order to produce 3D points from the laser distances as desired, a functional analysis leads to the definition of two functions to be realized by the on-board system (reported in Figure 2 as a data-flow diagram):

- Localization: need for precise, high-frequency and real-time position, speed and orientation of the vehicle;

---

<sup>\*</sup> Corresponding author.

- Geo-referencing: transforming the range data (distance and angle of the laser scanner) into 3D points expressed in a common geo-referenced frame.

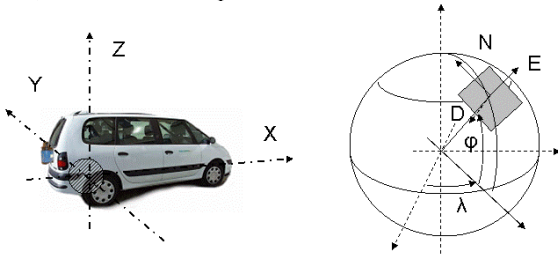


**Figure 2: framework for construction of 3D points**

To perform processing at the speed of acquisition, we need a specific implementation focussing on optimization of computation times. In the sub-sections 2.2 and 2.3, we explain the principles used for the two functions of localization and geo-referencing, in 2.4 the details of implementation, in 2.5 results obtained on two different outdoor scenes, and an analysis of the results in sub-section 2.6.

## 2.2 Vehicle localization

For precise and real-time localization we use a GPS-INS coupling (Abuhadrous et al. 2003a). Inertial Measurement Units (IMU) systems alone give only differential information (accelerations and rotation speeds), which need to be integrated in order to give position, orientation and speed of the vehicle (Inertial Navigation System, INS). These information are precise and high-rate yet subject to drift along time. GPS information give absolute position (and speed), but at lower rate, and they can sometimes be missing. A combination of both, by fusion of data, can be performed and give fast, accurate and absolute positions, orientations and speeds.

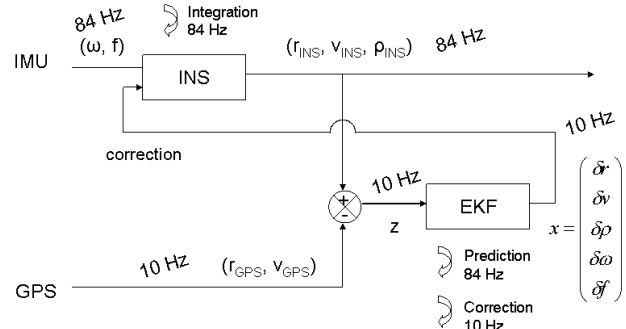


**Figure 3: reference frames**

The vehicle reference frame is defined with the X direction along the vehicle, Z pointing up, and its centre being located above the rear wheel axis (Figure 3). Its position is given in a reference frame fixed to the Earth, with the representation  $r$  (latitude, longitude, altitude). Its orientation and speed are given with respect to the local tangent plane (NED - North, East, Down). We use an additional reference frame  $\mathcal{R}^0$ , defined with the position and local tangent plane of the vehicle at some initial point, and we use for convenience the position  $s$  (X, Y, Z) of the vehicle in this frame. The orientation of the vehicle is expressed in NED frame with 3 Euler angles  $\rho$ , or equivalently with a rotation matrix  $R$ . The speed is denoted  $v$ .

To implement the Inertial Navigation System equations, we use the differential equations, relating accelerations  $f$  and angular speeds  $\omega$  to orientation  $\rho$ , speed  $v$  and position  $s$  of the vehicle (Farrell and Barth 1998). In order to perform fast computations, we make simplifying assumptions in the

equations (e.g. neglecting Coriolis terms that are low compared to the sensor sensibility, but expensive in time to compute). We use for numerical integration a first-order Euler scheme.



**Figure 4: GPS-INS coupling**

In order to correct the drift of INS, we use an Extended Kalman Filter (EKF) with feedback correction (Grewal et al. 2001), as represented on Figure 4. The observation of the filter is the vector of differences between GPS and INS positions and speeds. The state vector is the difference between actual values and INS values for position, speed, orientation, IMU values for angular speed and acceleration. Considering small variations around the actual values, a linearized equation can be written to perform prediction and correction steps of the filter. Prediction runs at the rate of INS computation, while correction runs when GPS data are available. When a correction has been computed, the IMU terms are fed forward as new IMU offsets in the INS equations. Predictions and corrections use noise models for IMU and GPS, which need numerical values determined through calibration or experimentation.

## 2.3 Geo-referencing of range data

The laser range data are given as distances and angles of measurement points in a vertical plane moving with the vehicle. It is possible to combine this data with the vehicle localization information, in order to produce sets of 3D points in a common terrestrial reference frame (Abuhadrous et al. 2003b). The frame change between range sensor and vehicle is constant, it needs to be determined through calibration. As the range data are given as a set of points, a simple interpolation is performed, to evaluate the exact position of the vehicle for a given point in the profile.

## 2.4 Implementation - the LARA-3D platform

We have implemented the two functions of localization and geo-referencing on a prototype vehicle, called LARA-3D. It consists of a car (Renault Espace) equipped with localization sensors (GPS, IMU), perception sensors (rotating laser range sensor, cameras), on-board PC computer with adapted software environment and data storage capacity.

For localization, we use a GPS antenna AgGPS132 from Trimble (AgGPS 2000), giving non differential absolute positions at a rate of 10 Hz, and a variable latency time comprised between 30 ms and 150 ms. The Inertial Measurement Unit is a strapdown VG600CA-200 from Crossbow (DMU 99), delivering data at a rate of 84 Hz. The laser range sensor is an LD Automotive from IBEO (LD IBEO), delivering 1080 distances per rotation at the rate of 10 rotations per second (scan frequency). On our platform we have an on-board computer with bi-processor Pentium III at 750 MHz using the Windows 2000 operating system.

To design the dedicated software, we use a specific environment for prototyping of real-time on-board applications, called <sup>RT</sup>MAPS (Intempora). It allows to design and implement processing algorithms in C++, and then test them on sets of real data acquired before, re-played at speeds lower (or faster) than the original ones. It allows an easy determination of computation times. This environment makes it possible to execute non optimized algorithms, measure computation times, and determine whether it is possible to speed up replay or if it is necessary to perform optimization. One considers that an algorithm supports real-time when a replay at normal speed allows processing without loss of data or synchronization. In that case it is possible to install the software on-board for simultaneous acquisition and processing. For the software prototyping, our PC was a laptop with Pentium M at 1.7 GHz, 1Mo level-2 cache memory, 1 Go RAM, graphics card NVidia GeForce 4 4200 GO with 64 Mo RAM, under Windows XP.

The two functions of localization and geo-referencing have been implemented as two <sup>RT</sup>MAPS components, which can be linked together in a data-flow interface. We also use components to decode raw input (sensors) data, which are in specific formats. For the localization function, we have added a first-order low-pass filter for IMU data, in order to eliminate high frequency noise. At the beginning of acquisition, we stay with the vehicle immobile a few seconds in order to determine the IMU initial offsets (biases). The initial orientation of the vehicle (boresight), needed for INS integration initialization, is determined with a compass. For the GPS, we take into account the latency time between position acquisition and delivery to correct the position; we also take into account and perform correction for a “lever-arm” effect, related to the distance between actual GPS antenna and centre of vehicle frame. For the Kalman filter, we give numerical values for the noise models: we use values coming from calibration (IMU) or experimentation (GPS). The frame change between laser and vehicle is determined with a hands-on procedure (measuring tape...).

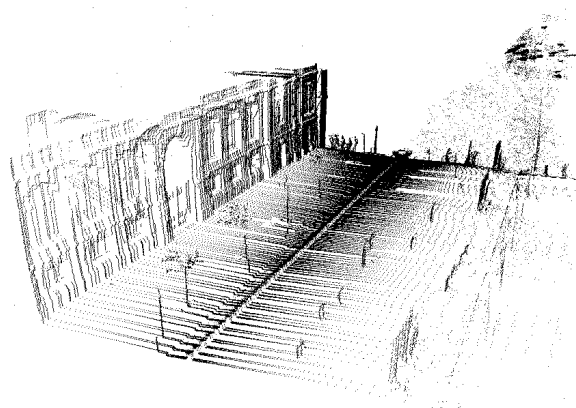
**2.5 Results**

We used the LARA-3D platform to perform acquisitions in various outdoors, urban and road, environments. We present in Table 1 a description of characteristics of these acquisitions. The PDOP signal (Position Dilution of Precision) is related to the accuracy of the GPS position (related to the number and positions of satellites), a lower PDOP yielding better accuracy.

**Table 1: Description of acquisitions**

Characteristics	Urban scene	Road scene
Distance	215 m	3 500 m
Acquisition time	2 min 07 s	11 min 50 s
<b>Speed (mean)</b>	<b>6,1 km / h</b>	<b>17,7 km / h</b>
Number of profiles	1 163	6 762
Number of points	720 840	7 606 898
Inter-profile mean dist.	18,5 cm	51,8 cm
Mean PDOP (GPS)	3,8	2,0

Figures 5 and 6 show the 3D points of the two scenes. In the urban scene (Figure 5), one may distinguish details such as trees and even pedestrians. The road scene (Figure 6) consists of a test track for vehicle, 3.5 km long. Our system can perform acquisition at low or normal speed, but in both acquisitions presented here a slow pace was used in order to have a high density of points.



**Figure 5: Urban scene**



**Figure 6: Complete road scene (bird view)**

**2.6 Analysis of the results**

The system developed gives good apparent (qualitative) results. In this section, we assess its real-time performance, and the precision of the data measured.

**2.6.1 Real-time performance**

We aim at developing an integrated on-board system for on-the-way digitizing and modelling. This means that the data can be processed in real-time, at the acquisition speed. The notion of “real-time” used here, means that computations shall be performed fast enough. We want to assess in the following these computation times for the system. In order to do so, we performed several measurements, with various replay speeds. The original acquisition speed is called real-time speed; the replay speed is expressed in percent of the original one, written for example 30% RT for a speed at 30% of the original one.

The two functions of localization and geo-referencing are implemented as 4 iterative loops, which run at different rates. For localization, there are 3 loops: INS at 84 Hz, EKF prediction at 84 Hz and EKF correction at 10 Hz or less; for geo-referencing, one loop at 10 Hz. For each one of these loops is calculated an equivalent CPU load, as the product of the mean computation time of the loop during the whole acquisition process, and its rate. It is expressed in percent (%) and gives information on how much of the processor is used or would be used at normal replay speed. An equivalent CPU load can then be determined for the complete process as the sum of the loads for the loops. If the equivalent CPU load is close to 100% or above, the software still runs but one may see bad synchronization or loss of data. We can also calculate the exact number of data lost and report it.

In Table 2 are presented the computation times, data loss and equivalent CPU load, determined for the urban scene in two

configurations: left column, a replay speed of 30% RT, and right column, a replay speed of 100% RT with higher priorities for the components threads. We observe in the right column that we succeed in performing real-time performance with no loss of data and an actual CPU load below 100% at a normal replay speed; we notice the importance and direct impact of tuning correctly the threads priorities, with low priorities even at a lower speed of replay the performances are not as good and we observe loss of data.

**Table 2: Computation times, CPU load, data loss**

Replay speed		30% RT	100% RT, priorities
Time (ms)	CPU load (%)		
INS 84 Hz	mean	0.12 ms	0.06 ms
	CPU	1.0 %	0.5 %
EKF prediction 84 Hz	mean	6.43 ms	5.30 ms
	CPU	54.0 %	44.5 %
EKF correction 10 Hz	mean	6.33 ms	5.42 ms
	CPU	6.3 %	5.4 %
<b>Total localization</b>	<b>CPU</b>	<b>61.3 %</b>	<b>50.4 %</b>
<b>Geo-referencing</b> 10 Hz	mean	14.57 ms	12.09 ms
	CPU	14.6 %	<b>12.1 %</b>
<b>Total data lost</b>		<b>0.6 %</b>	<b>none</b>
<b>Total equiv. CPU load</b>		<b>75.9 %</b>	<b>62.5 %</b>

### 2.6.2 Precision

One has to make the distinction between absolute and relative precision of data. The absolute precision is related to systematic bias in the values, while the relative precision is related to random noise affecting the data.

In our case, we have several sources for absolute and relative precision. For the localization function, the best relative precision is given by the INS data, the noise coming from the IMU noise; the INS gives bad absolute precision, because of a drift that is corrected by the GPS; hence the best absolute precision is given by the GPS (however not very precise, because of the use of simple GPS without corrections such as DGPS, RTK...). In a series of outdoor experiments on the road scene (track), we have compared our results with reference measurements, and we can give rough estimates of the precision of our GPS-INS localization system: for relative precision, standard deviation  $\sigma \approx 1$  mm, and absolute precision (bias) less than 4 m.

**Table 3: Estimated precision of the 3D points**

Precision	Relative (std deviation $\sigma$ )	Absolute (bias)
Points 3D	$\approx 5$ cm (mainly in laser dir.)	< 4m (when GPS available)

We have verified on a workbench the correct calibration of the range sensor, and the standard deviation of 5 cm in distance announced by the manufacturer. In the geo-referencing, additional sources of inaccuracy are in laser-vehicle calibration, and propagation of the vehicle orientation random noise coming from the EKF. Calculations show that, for the distances considered in our experiments, the order of magnitude of the effect of the orientation noise on the geo-referenced 3D points is more than 10 times lower than the range noise. Hence we can consider that the random noise for the positions of 3D points is mainly in the laser direction, with the magnitude of the range random noise. Concerning laser-vehicle calibration, we assume

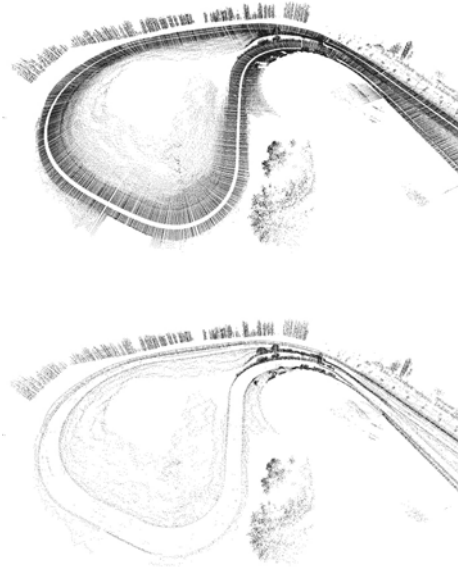
that a correct calibration has been done. For absolute precision, the remaining element is the absolute precision of position, coming from the GPS. This leads to Table 3 which summarizes the absolute and relative precision of the 3D points delivered by the LARA-3D system.

## 3. CITY AND ROAD MODELLING

We present in this section algorithms for city and road modelling, designed for real-time performances. For this reason, they process directly data profiles, as soon as they are available. In Section 3.1 we present adaptive decimation to reduce the number of points, in Section 3.2, segmentation for urban modelling, and in Section 3.3 road features extraction.

### 3.1 Geometric adaptive decimation

The LARA-3D system produces high quantities of points, even for short acquisition distances, which can make display or interactive visualization difficult. For this reason we need to perform data decimation, in order to reduce the number of points. We have adapted a simple iterative algorithm, processing data profile per profile and performing polygonal approximation with only two thresholds to set. The principle of the algorithm is as follows: we start with a first segment linking the first and last points of a profile; we perform subdivision until the mean distance of points to their corresponding segment is below a threshold, and the distance of each point to its segment is below a second threshold. This algorithm, very simple, gives good results and is very cheap in computation time (Figure 7, Table 4). Its interest is that it keeps points where they are geometrically meaningful; this is why it is called geometric adaptive decimation.



**Figure 7: decimation over the road scene**

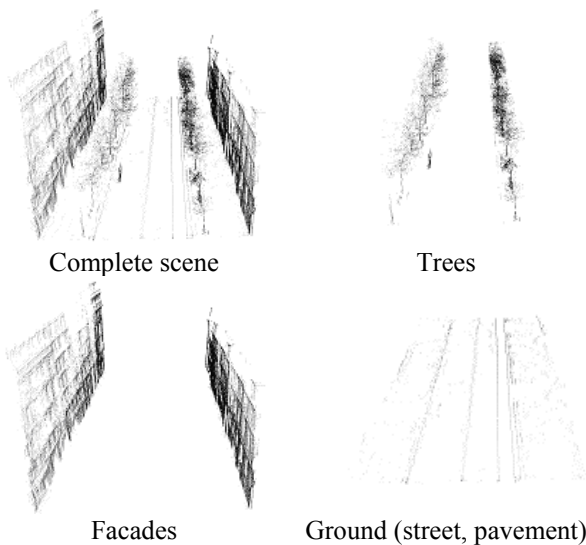
**Table 4 : decimation algorithm performances**

Characteristics		Urban scene	Road scene
Initial nb. of points		720 840	7 606 898
Nb. pts. after decim.		120 678	276 337
Decimation rate		17 %	4 %
decimation 10 Hz	Time	1.3 ms	0.7 ms
	CPU	1.3 %	0.7 %

### 3.2 Segmentation for city modelling

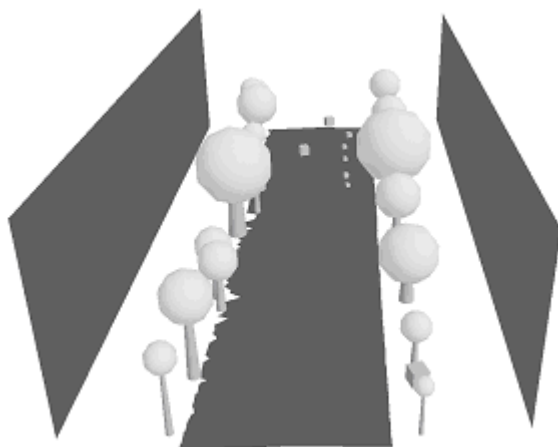
We have developed an algorithm to perform segmentation and modelling in real-time, based on the profiles of points, for the following elements in the scene: ground (road and pavement); facades; trees; unknown or unidentified (pedestrians, cars, street lights, artefacts) (Ammoun et al. 2004). Segmentation is performed only on the geometry of the scene, considering the following characteristics:

- Ground and road: horizontal plane with high density of points
- Facades: vertical planes
- Trees: free shapes whose projections onto the ground have a specific width



**Figure 8: urban scene, original and segmented**

The ground is detected and segmented first, using a histogram of vertical coordinates of points, which shows a peak at the ground level. The points corresponding to this peak and their neighbours in the profile are segmented as ground. The road is extracted from the “ground” class, using the geometric decimation algorithm.



**Figure 9: Urban scene modelling**

For facades and trees, we use a histogram of horizontal coordinates of the profile, from which the ground points have been taken away. Such histogram shows disconnected elements, some belonging to facades, trees, or unidentified objects. The

study of continuity of these elements along several consecutives profiles, with some probability reasoning, allows labelling of the object. On Figure 8 are represented a complete scene, segmented trees, facades and (decimated) ground.

Once the points are segmented into objects, and the objects labelled, one can model the objects. We performed modelling with the following models:

- Road: triangulation using the points at the border and the centre of the road;
- Facades: vertical plane, envelope of the points;
- Trees: sphere for foliage, cone for the trunk, parameters adapted to the points (sphere: radius based on the horizontal projection of the tree points on the ground, centre at the centre of gravity of the tree points; cone: height equal to the tree height, ground radius equal to  $\frac{1}{4}$  of the sphere radius);
- Others: cube envelope of the points.

Such modelling can also be performed on-line. Results on the urban scene are shown on Figure 9.

The computation times for the loops of segmentation and modelling are given in Table 5, showing real-time performances for these algorithms.

**Table 5: computation times and CPU loads for segmentation and modelling – urban scene**

Segmentation 10 Hz	mean	5.8 ms
	CPU	5.8 %
Modelling 10 Hz	mean	11.9 ms
	CPU	11.9 %

### 3.3 Road features extraction

The 3D road models that LARA-3D can produce make it possible to extract useful information: width, curvature, slope, banking, and others. We have developed computation of width and curvature (Ammoun et al. 2004).

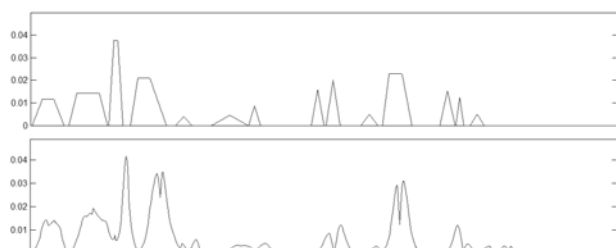
The decimation algorithm gives easily, as a by-product, the borders of the roads. One can see on Figure 10 a comparison of borders obtained by this method (left) and actual borders. Artefacts with our method are due to imprecise geometric borders, like road entrance or parking area. From the borders, it is then straightforward to compute the width of the road. The comparison shows qualitatively effective results, the computation time being the same as for decimation (mean computation time of 0.7 ms).



**Figure 10: road borders, comp. (left), actual (right)**

For curvature, we consider a third-order polynomial fitting on the horizontal projections of the road borders, for 50 consecutive points, and we determine the curvature from analytic formula. This method has been tested off-line. We present on Figure 11 a comparison between theoretic curvature of a precise road model, and the curvature computed by this method, represented along a curvilinear abscissa. This figure shows good qualitative performance for curvature determination, with how-

ever some artefacts and discontinuities that would need to be filtered.



**Figure 11: curvatures of the road, theoretic (above) and computed (below)**

#### 4. CONCLUSION

We have presented an integrated on-board laser range sensing system for on-the-way city and road modelling. Qualitative results show dense and realistic modelling on the two kinds of environments addressed (city and road) with a good quality in details. A precise analysis has validated that our implementation supports real-time (processing time below acquisition rate) in normal operating conditions, and has presented the precision of the 3D points obtained. Additional modelling can be performed on the data on-line, such as adaptive decimation, segmentation, or road feature extraction.

The system presented can be used for the scanning of large areas accessible to a car, at its level and normal speed, which was our initial goal. In our experiment, to acquire and process 3.5 km of data (road scene), we needed at a speed of 17.7 km/h, giving a high resolution with a distance of 51.8 cm between profiles (1 080 points per profile), a total acquisition and processing time of 11'50".

Several improvements could be made to the system, e.g. for better localization with DGPS or additional sensors, or for more realistic modelling with the addition of texture in the same real-time conditions.

#### REFERENCES

- Ammoun S., Abuhadrous I., Nashashibi F., Goulette F., Laurgeau C., 2004. Modélisation 3D d'environnements urbains et routiers numérisés par télémétrie laser embarqué. *In Proc. Numérisation 3D-Scanning 2004, Paris – Harbour Conferences*.
- Abuhadrous I., Nashashibi F., Goulette F., Laurgeau C., 2003(a). 3-D Land Vehicle Localization: a Real-time Multi-Sensor Data Fusion Approach using <sup>RT</sup>MAPS. *In Proc. of the 11<sup>th</sup> International Conference on Advanced Robotics*. June 30<sup>th</sup> – July 3<sup>rd</sup>, 2003, University of Coimbra, Portugal.
- Abuhadrous I., Nashashibi F., Goulette F., Laurgeau C., 2003(b). Onboard Real-time system for Digitizing and Georeferencing of 3D Urban Environments. *In Proc. of the 11<sup>th</sup> International Conference on Advanced Robotics*. June 30<sup>th</sup> – July 3<sup>rd</sup> 2003, University of Coimbra, Portugal.
- (AgGPS 2000) AgGPS 124/132 Operation Manual, 2000. Trimble Navigation Limited, USA 2000, <http://www.trimble.com/>.
- Allen P.K., Stamos I., Gueorguiev A., Gold E. and Blaer P., 2001. AVENUE: Automated Site Modeling in Urban Environments. *In Proc. of the 3rd Conference on Digital Imaging and Modeling (3DIM'01)*, Quebec City, Canada, May 2001, pp.357-364.
- (DMU 99) DMU User's Manual, 1999. DMU-6X/ DMU-VGX/ DMU-FOG. *Crossbow Technology Inc., USA*, <http://www.xbow.com/>.
- Farrell J.A. and Barth M., 1998. The Global Positioning System and Inertial Navigation, *MacGraw-Hill, 1998*.
- Früh C. and Zakhor A., 2001. Fast 3D Model Generation in Urban Environment. *Int. Conf. Multisensor Fusion and Integration for Intelligent Systems*, Baden-Baden, Germany, August 2001.
- Früh C. and Zakhor A., 2003. Constructing 3D City Models by Merging Aerial and Ground Views. *IEEE Computer Graphics and Applications*, nov.-dec.2003, pp.52-61.
- Früh C. and Zakhor A., 2004. An automated method for large scale, ground based city model acquisition. *Int. J. Computer Vision*, vol.60, no.1, Oct.2004, pp.5-24.
- Grewal M.S., Weill L.R. and Andrews A.P., 2001. Kalman Filtering: Theory and Practice (Using MATLAB), 2<sup>nd</sup> edition, *Wiley and Sons, 2001*.
- (Intempora) <sup>RT</sup>MAPS User's Manual. *Intempora SA*, <http://www.intempora.com>.
- (LD IBEO) IBEO LD Automotive, <http://www.ibeo.de>
- Stamos I., Allen P.K., 2000. 3D Model Construction using range and Image data. *In Proc Int. Conf. on Computer Vision and Pattern Recognition (CVPR) 2000*.
- Zhao H., Shibasaki R., 2001. Reconstructing Urban 3D Model using Vehicle-Borne Laser Range Scanners. *Proc. 3rd Int. Conf. on 3D Digital Imaging and Modeling (3DIM)*, May 2001, Québec City, Canada.
- Zhao H., Shibasaki R., 2003a. A Vehicle-borne Urban 3D Acquisition System using Single-row Laser Range Scanners, *IEEE Trans. SMC Part B: Cybernetics*, vol.33, no. 4, August 2003.
- Zhao H., Shibasaki R., 2003b. Special Issue on Computer Vision System : Reconstructing Textured CAD Model of Urban Environment using Vehicle-borne Laser Range Scanners and Line Cameras. *Machine Vision and Applications*, 14 (2003) 1, 35-41.