Loop closure detection using small-sized signatures from 3D LIDAR data

Naveed Muhammad[1,2]
naveed.muhammad@laas.fr

Simon Lacroix[1,2]
simon.lacroix@laas.fr

***Abstract*— This article presents a technique for loop closure detection using lidar data for autonomous mobile robots. The technique consists in extracting, indexing and matching a set of small-sized signatures from lidar data. These signatures are based on histograms of local surface normals for 3D point clouds. As the robot moves, some histograms are automatically selected as key-histograms and histograms of newly acquired lidar scans are matched to the key-histograms in order to detect loop-closures. Results with real 3D lidar data validate the proposed technique.**

Keywords: *Loop closure, Velodyne, lidar, Histogram*

I. INTRODUCTION

An essential property to endow a mobile robot with autonomy is its ability localize to itself in an environment which is a priori unknown. This is the reason why a vast amount of work has been done on robot localization as well on simultaneous localization and mapping (SLAM) during last couple of decades. The SLAM problem has been extensively studied and proposed solutions range from extended Kalman filtering (EKF) to particle filtering and to biologically inspired techniques like RatSLAM [1].

Originally SLAM was mostly performed using single-beam planar laser scanners and during the last decade a lot of work has been done on SLAM using vision. Recently the introduction of more powerful and fast multi-beam laser scanners (such as Velodyne HDL-64E S2 [2]) and their successful exploitation in DARPA autonomous vehicle challenges has proven their potential and scope to be employed in robotics.

One important aspect of a SLAM system is the "loop closure" problem. A loop closure occurs when a robot while creating a map of its environment re-visits an already visited location: it detects that it has already visited the locations, and then properly incorporates this loop closure into the map that it is building. Within the SLAM framework, loop closures are detected by matching mapped landmarks in the environment: this process requires very discriminative landmark signatures and can hardly be driven by the current position estimate for long range trajectories, when the current estimate become very uncertain. Loop closures are especially important in graph-based SLAM techniques (such as [3]) because loop closures are essential to optimize robot pose estimates in such methods. In this paper we consider a SLAM independant loop closure detection process, that relies only on signatures computed from panoramic lidar scans.

The multi-beam laser scanner Velodyne HDL-64E S2 consists of 64 lasers located on a spinning head which can spin at a rate of 5 to 15 Hz, and provides 3D data (27° x 360° field of view) about its surroundings at a rate of 1.33 million points per second. The huge data rate makes it infeasible to store and match the whole point clouds captured by the device and it is desirable to extract some small-sized signatures from each 360° scan that can be used to perform loop closure detection to facilitate a SLAM system. [4] present a method for automated extraction of planes and calculation of transformation parameters between 3D laser scans using the matching planar patches from the scans. The method uses region growing for plane extraction which is computationally very expensive and especially for high density 3D data captured at fast rates such as from the Velodyne. Our approach extracts histogram-based signatures from 3D lidar data and uses them for loop closure detection. The histograms are based on local surface normals for each 360° scan acquired by the lidar device as the robot moves.

The paper is organized as follows: section II presents some related work, including histogram-based signature extraction from panoramic images, section III details different steps involved in the proposed technique *i.e.* extracting local surface normals for lidar data, computing histograms and histogram matching for loop closure detection, and section IV presents some results on real lidar data.

II. RELATED WORK

In [5], the authors present a technique for robot localization and loop closure using panoramic images. Panoramic images are split into three rings, the smallest and largest ring correspond to areas nearer and farther from the robot respectively. The reason for splitting up the image into rings is that in panoramic images, the scene close to the sensor (represented by smaller rings) changes much quicker than the scene farther from the sensor (represented by larger rings). The authors define eight Gaussian-derivative based local characteristics. For each ring of each panoramic image, histograms are computed for each of the eight defined characteristics. As the robot moves, a histogram is saved as a "key histogram" if it has a significant difference compared to last key histogram (defined by a threshold value). This is

1: CNRS; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France
2: Université de Toulouse ; UPS , INSA , INP, ISAE ; LAAS ; F-31077 Toulouse, France

Fig. 1. Velodyne HDL-64E S2 fitted on the robot Robufast, along with GPS and other sensors

done because saving the histograms for each frame captured by the sensor in not necessary. During the localization phase, acquired histograms are matched with previously saved histograms in order to detect loop closures. The difference between histograms can be computed using different measures including histogram intersection, Euclidian, Quadratic, Mahalanobis, and Haussler's distances or χ^2 statistic. Authors have found χ^2 statistic to perform the best. Computing and matching histograms on rings of panoramic images makes the process rotation invariant. During the localization phase, once the histogram closest to that of current location is found, relative orientation of the robot can be estimated by discretizing the rings into bins and computing the correlation between gray values of the discretized rings.

The above technique can be extended to lidar data from Velodyne and similar devices because a 360° scan for such devices is analogous to a panoramic image. This histogram based technique can lead to a "loop closure detection" which can then be proceeded by an actual loop closure in feature space. These features can be point, line or plane based. An interesting and innovative work about extraction of interest points from lidar data has been presented in [6] and [7], where the Kanade-Tomasi corner detector on 2D and 3D lidar data is applied by projecting the data into 2D images. 2D lidar data is converted into an image by convolving it with a Gaussian kernel whose width is adjusted according to sensor range noise as well as positional uncertainty of each scan point arising from scan sampling/discretization. One advantage of this Gaussian smoothing is that at short ranges it smoothes the scanned surface which might otherwise look noisy as a result of range-measurement error. In order to rasterize 3D lidar data, the authors discretize the scan into a 2D grid (in horizontal plane). Then in each cell, the maximum and minimum height of present points is checked and their difference is calculated. An image is then drawn based on this height difference value for each grid cell, and a corner detection is applied to the image. The authors present results in both natural and indoor environments for 2D data and outdoor 3D data (from Velodyne lidar) and the method appears to perform well. In [8], the authors propose the extension of Harris detector to 3D. [9] presents a comparison of line extraction algorithms for 2D Lidar data. These algorithms include Split and merge, incremental algorithm, RANSAC and Hough-transform etc. But the implementation of these techniques on noisy 3D lidar data such as from the Velodyne is yet to be investigated to the best of our knowledge. Alternatively, the loop closure detection step can also be achieved by performing iterative closet point algorithm or one of it's variants such as Generalized-ICP presented in [10] to get a more precise transformation between the two robot locations for which a loop closure has been detected – but this is a costly solution.

III. Description of the Proposed Approach

Our approach relies on the computation of very small-sized signatures extracted from lidar data. These signatures are based on local surface normals in the point cloud gathered by the lidar. The technique can be summarized as extraction of local surface normals, computing their histograms, and computing difference between these histograms to detect loop closures. The details on each of these steps are presented below.

A. Local surface normal extraction

Running at 5 Hz, each 360° scan provided by our lidar sensor consists of around 266,000 points. We discard the points whose distance less than 3m and greater than 50m from the sensor. The reason for discarding points less than 3m away from the sensor is that practically speaking most of these points are returns from ground when the robot is in an outdoor setting. The reason for discarding points beyond 50m range is that the working range of the sensor for low reflectivity objects is 50m as mentioned in the data sheet.

[11] presents a method to compute local surface normals from high resolution (Velodyne) lidar data. For any given point in the point cloud, its 20 nearest neighbors are selected from the whole point cloud and a plane fitting is performed to find the local surface normal at the point in question. [12] present another method of approximating local surface normals by exploiting the geometry of laser beam placement inside the Velodyne lidar device. For a given point, its right and left neighbors are the points acquired by the same laser beam immediately before and after the point in question. Upper and lower neighbors are the nearest points acquired by laser beams which are immediately above and below (*i.e.* having higher and lower pitch angles respectively) the laser beam that acquired the point in question. Once the four neighbors have been chosen, the local surface normal is estimated by taking the cross products between displacement vectors formed by the given point and the four neighbors, and then geometrically averaging the resulting cross products.

We implemented both methods, but in the current implementation and presented results the method from [12] is used because it is much faster. To make the four vectors have comparable magnitudes (and thus reducing the effect of noise in range data on surface normal computation), for the right and left neighbors we take the 5th point acquired before and after any given point instead of points acquired

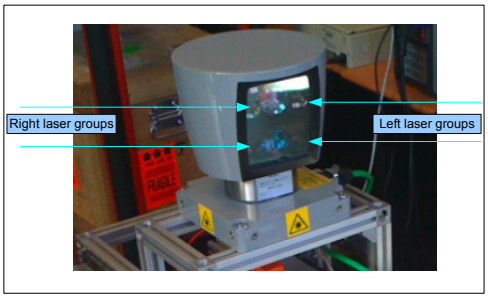

Fig. 2. Velodyne HDL 64E-S2

immediately before and after the given point. Similarly for upper and lower neighbors, we take the closest points from the second laser beam above and second laser beam below the laser beam instead of taking the closest points from the laser beams immediately above and below. In fact in the Velodyne HDL 64E-S2 device, 64 laser beams are distributed into 4 groups of 16 lasers each (*c.f.* fig. 2). Two groups are located on the right side on device's spinning head and two are located on the left. There is a slight misalignment in the points captured by the laser-groups located on the right and left hand sides of the device. This problem has also been pointed out in [13] and is a result of inaccurate calibration. Lasers beams two steps above and below any laser beam are in fact from the laser-group located on same side (*i.e.* right or left) of the lidar head. This fact makes the points from laser beams two steps above and below the point in question the best candidates for upper and lower neighbors rather than the points from laser beams immediately above and below the point in question. The surface normal computation process is shown in fig. 3. Four neighbors of the point at which the surface normal is to be computed are shown circled in red. The cross products formed by the considered point in question and the four neighbors are shown in blue while the geometric average of these cross products is shown as the red line. This red line is hence the estimated surface normal at the point in question.

Fig. 4 shows the local surface normals extracted for data acquired by a subset of lasers (10 out of 64) for scan no. 460 (*c.f.* fig. 8) of our data set.

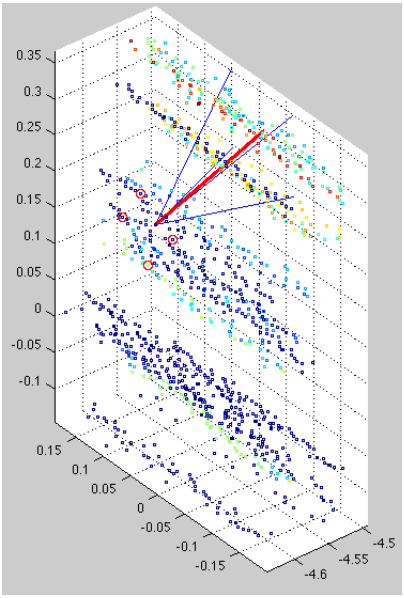

Fig. 3. Local surface normal computation for a point: Four neighbors of the considered point are circled in red. Four neighborhood cross products are represented by four blue lines and their geometric average (which is also the local surface normal estimate) is shown as the red line.

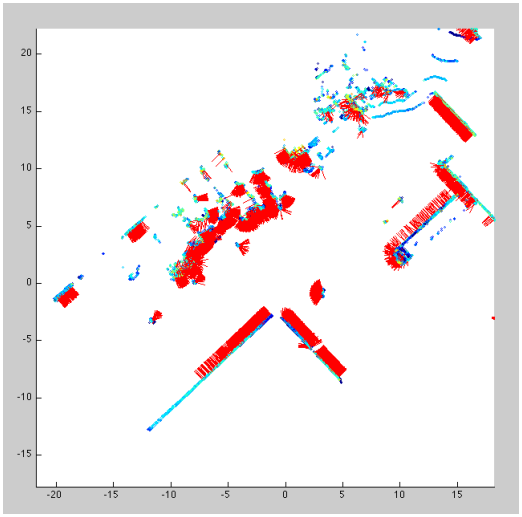

Fig. 4. Top view of one 360° scan with local surface normals (shown as red lines). Thick red areas indicate the presence of walls in robot's environment. Units in meters.

B. Histogram based signature extraction

Once we have the local surface normals, we take the dot products of these normal vectors with the vertical $\hat{z}$. We assume that the robot is moving on a sufficiently flat terrain so the $z-axis$ of the $sensor-coordinate frame$ is considered to be the vertical. In case the terrain is not flat, the robot roll and pitch information can be used to transform the 3D scans into a $global-coordinate frame$ and then the signature extraction can be performed. Taking this dot product is in fact one way of quantifying the structure in the environment around the robot. The value of these dot products ranges between -1 and 1. Normal vectors that are parallel to the $x-y$ plane result in the dot product having low or nearly zero values whereas the normal vectors parallel to $z-axis$ result in dot products having values near -1 and 1. We discretize this range between -1 and 1 into 101 bins for histogram computation.

C. Histogram distance

In order to perform histogram based loop closure detection, a quantitative measure of similarity/dissimilarity between any two histograms (corresponding to lidar scans acquired at two locations in the data set) is required. If the histogram distance between two histograms is less than a threshold value a loop-closure can be registered and vice versa. [14] gives an overview of different distance and similarity measures between two histograms. Among the many

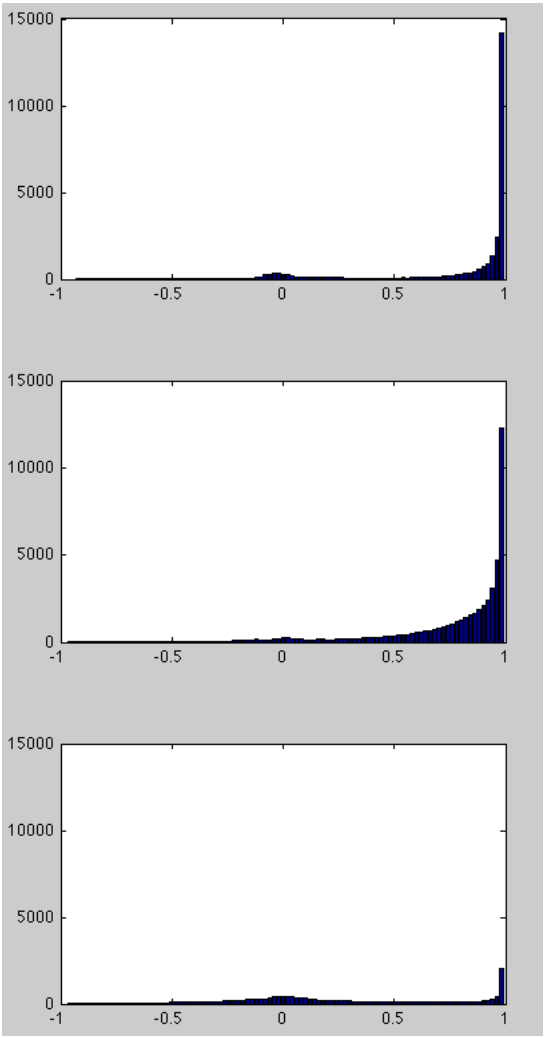

Fig. 5. Histograms based on $normal-vectors.\hat{z}$ for three (1^{st}, 1200^{th} and 1320^{th}) 360° scans of our experimentation data set

TABLE I

HISTOGRAM DISTANCES BETWEEN EXAMPLE SCANS OF FIG. 5

Scans	χ^2 dist.	Sørensen dist.
1, 1200	14510.81	0.351
1, 1320	17107.70	0.579
1200, 1320	32038.13	0.654

possible candidates for distance measurement between two histograms, we have chosen the χ^2 and Sørensen measures for our implementation. These two measures were chosen because they were found to be giving consistent distance values for different similar and dissimilar histograms. The two histograms measures are defined as follows:

$$D_{\chi^2} = \sum_{i=1}^{d}((P_i - Q_i)^2/(P_i + Q_i + 1)) \quad (1)$$

$$D_{Sørensen} = \sum_{i=1}^{d}|P_i - Q_i| / \sum_{i=1}^{d}(P_i + Q_i) \quad (2)$$

where P and Q are the two histograms and d is the number of histogram bins.

Fig. 5 shows the histograms for three 360° scans from our experimentation data set. The high value at bins near 1 corresponds to the ground points having surface normals along positive $z-axis$. The number of ground points can vary a lot depending on the "emptiness" of the robot environment (as apparent in the example histograms) and thus ground data is as important and handy as other points form the environment to characterize robot position. Table I shows the histogram distances between the three example histograms of fig. 5.

D. Key-histogram selection

If the acquisition rate of the lidar device at hand is fast enough, the difference between two consecutive scans acquired by the device is slight. In this situation storing every 3D scan or its histogram and comparing it to every single previously acquired scan or histogram is not required. For instance the Velodyne HDL 64E-S2 lidar device can be programmed to rotate at the speeds between 5-15 Hz. Our device is programmed to run at the minimum speed *i.e.* 5 Hz which means if the robot runs at a speed of 1m/sec, the distance at with the two scans are acquired is only 20 cm.

When the robot starts to move, the first acquired scan is stored as the first key-histogram. As the robot continues and new scans are acquired, the distance between each new scan and the last key histogram is computed. A new scan and its corresponding histogram is stored as a key-scan and a key-histogram if the distance between the last key-histogram and the current one is more than an empirical threshold T_{key}, and the process continues.

E. Loop closure detection

As the robot moves, each new histogram (from the most recently acquired scan) is matched with previously stored key-histograms. Lets denote the current histogram for the scan acquired at time t by H_{curr} and each of the previously stored key-histograms by H_i where i ranges from 1 to n, n being the total number of key-histograms stored up till time t. We compute the χ^2 and Sørensen distances between H_{curr} and H_i for i ranging from 1 to $n-m$ where m is a number which limits the histogram matching to the scans that were acquired sufficiently earlier than the current scan. In our experimentation we have set the value of m to 15. This value of m is just a matter of choice and smaller values of m might result in loop closure detection for the locations that the robot recently visited and therefore might not be of interest. For instance if the value of m is set to 2 and the robot is moving in a straight line at a slow speed, the loop closure detection algorithm would signal the detection of a loop closure just because current histogram would match the key-histograms very recently acquired, where as in reality this is not a real loop closure because the robot has not even left this area.

During the histogram matching step, if both χ^2 and Sørensen distances between histograms H_{curr} and H_i are smaller than the two threshold values T_{χ^2} and $T_{Sør}$ a "loop

Fig. 6. Robot trajectory in our test environment

closure candidate" is registered. If this occurs at multiple key-locations i, all of them are registered as loop closure candidates. The location i which has the minimum Sørensen distance among all the loop closure candidates registered is taken as the loop closure detection location for the current scan.

IV. Results

We implemented the technique on a data set gathered by a Velodyne sensor fitted on robot Robufast (*c.f.* fig. 1). The implementation was done offline in Matlab. The data set consists of 1505 360° lidar scans which correspond to nearly five minutes of robot motion and data acquisition. We also have the centimeter accuracy D-GPS data for corresponding scans to be used as ground truth for validation of the technique. Fig. 6 shows the corresponding robot trajectory in our test environment.

In our experiments the threshold values T_{key}, T_{χ^2} and $T_{Sør}$ were set to 260, 434 and 0.0391 respectively. Fig. 7 shows the histograms distance evolution keeping scan no. 1 as the reference scan. Solid red lines represent T_{χ^2} and $T_{Sør}$. Low histogram distance values for scans around 1500 indicate the presence of a loop closure as the robot start and end points for the 1505-scan long sequence are very close to each other (*c.f.* fig. 6).

A total of 397 key-histograms were selected out of the total 1505 scans. Fig. 8 shows the loop closures detected for the experimental data set. Loop closures are shown by thick red lines between current robot position and the location where loop closure was detected.

Results also show a false positive detection between robot locations around 950 and 540. This is a result of very similar structure in robot's environment at these locations. The two locations are shown as blue and green spots in fig. 6. The features in the environment that make these two locations seem so similar to the robot are the corners and planar walls marked as "a" and "b" in blue and green (as perceived from

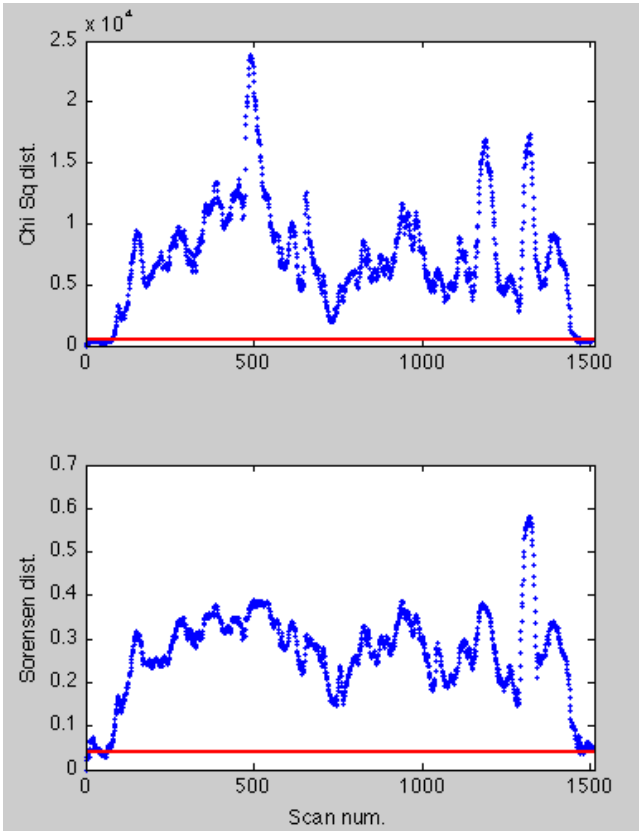

Fig. 7. Evolution of histogram distance measures for scan no. 1 as reference (blue), corresponding threshold (red). The histogram distance values for the scans at the end of the sequence are very low indicating a possible loop closure.

blue and green spots respectively). The histograms based signatures used in the implementation are rotation invariant. One way to get rid of possible false loop closure detections is to discretize the two candidate 360° scans into bins in circular sense and calculating the correlation between them (as in [15]) to accept or discard the possible loop closure detection. This correlation test can also lead to the estimation of relative robot heading between the candidate scans in case of a true loop closure detection.

As mentioned in section II, once a loop closure is detected, a Generalized-ICP process can be applied [10] to get a more precise transformation between two robot locations for which a loop closure has been detected. Matrix $T_{918-329}$ below shows the homogeneous transformation obtained by applying Generalized-ICP for the loop-closure detected between positions 918 and 329 (the initial transformation used for performing this step was the identity), estimated with the Generalized-ICP implementation published online by [10].

$$T_{918-329} = \begin{bmatrix} 0.942 & 0.336 & -0.016 & 5.006 \\ -0.336 & 0.942 & -0.007 & 2.372 \\ 0.013 & 0.012 & 1.000 & 0.063 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$$

Matrix $T_{1505-63}$ shows GICP results for the loop closure detected between start and finish locations of the robot. Note

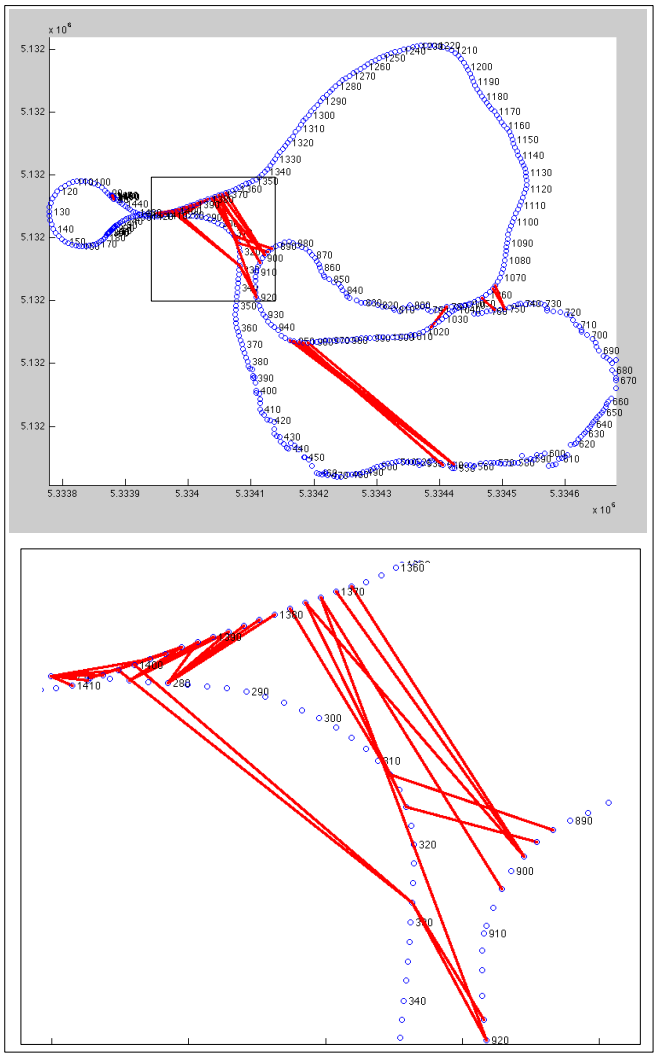

Fig. 8. Loop closure detections shown as red lines along the robot trajectory: whole trajectory (top), zoomed-in view (bottom).

the small values (in the order of a few centimeters) in the translation part of the transformation matrix representing the close vicinity of the two positions at which the scans were captured.

$$T_{1505-63} = \begin{bmatrix} 0.997 & 0.083 & 0.003 & 0.101 \\ -0.083 & 0.997 & -0.003 & -0.045 \\ -0.004 & 0.002 & 1.000 & 0.006 \\ 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$$

V. Conclusions

The ability to successfully detect and perform loop closures is an important requirement for any robot intending to move around autonomously in an a-priori unknown environment. In this paper we have presented a technique for loop closure detection, based on the extraction of small-sized histogram-based signatures from lidar data. Small-sized signatures are especially interesting when working with novel lidar sensors (such as the Velodyne HDL 64E-S2) that produce huge amounts of 3D data in short times and storing and processing all the 3D data in its raw form requires a lot of memory and computation. We have presented the results validating the proposed technique on a 5 minutes long data set captured by the Velodyne sensor in a semi-urban outdoor setting.

We intend to implement the technique onboard the robot to validate it in real-time, and to work on enhancing the defined signatures by using not only the surface-normal-based histograms but also more information from 3D lidar data in order to make the technique more robust.

Acknowledgments

This work has been partially funded by the Action project (action.onera.fr).

References

[1] M. Milford, G. Wyeth, and D. Prasser, "Ratslam: a hippocampal model for simultaneous localization and mapping," in *IEEE International Conference on Robotics and Automation*, vol. 1, 2004, pp. 403–408.

[2] *HDL 64E-S2 User's Manual – www.velodyne.com/lidar/*, Velodyne Lidar Inc, 345 Digital Drive, Morgan Hill, CA 95037, 2011.

[3] S. Thrun and M. Montemerlo, "The graphslam algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, May-June 2006.

[4] C. Dold and C. Brenner, "registration of terrestrial laser scanning data using planar patches and image data," in *ISPRS Commission V Symposium 'Image Engineering and Vision Metrology'*, September 2006.

[5] J.-J. Gonzalez-Barbosa and S. Lacroix, "Rover localization in natural environments by indexing panoramic images," in *IEEE International Conference on Robotics and Automation*, Washington DC, USA, May 2002.

[6] Y. Li and E. Olson, "Extracting general-purpose features from lidar data," in *IEEE International Conference on Robotics and Automation*, May 2010.

[7] Y. Li and E. B. Olson, "A general purpose feature extractor for light detection and ranging data," *Sensors*, 2010.

[8] F. Viksten, K. Nordberg, and M. Kalms, "Point-of-interest detection for range data," in *IEEE International Conference on Pattern Recognition*, Dec 2008.

[9] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2d range data for indoor mobile robotics," *Autonomous Robots*, vol. 23, pp. 97–111, 2007.

[10] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: Science and Systems*, 2009.

[11] J. Levinson and S. Thrun, "Unsupervised calibration for multi-beam lasers," in *12th International Symposium on Experimental Robotics*, December 2010.

[12] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion," in *IEEE Intelligent Vehicles Symposium*, June 2009, http://homepages.laas.fr/nmuhamma/intranet/MOOSMANN-IVS-2009.pdf.

[13] N. Muhammad and S. Lacroix, "Calibration of a rotating multi-beam lidar," in *IEEE International Conference on Intelligent Robots and Systems*, 2010.

[14] S.-H. Cha, "Taxonomy of nominal type histogram distance measures," in *American Conference on Applied Mathematics*. World Scientific and Engineering Academy and Society (WSEAS), 2008, pp. 325–330.

[15] J.-J. Gonzalez-Barbosa, "Vision panoromique pour la robotique mobile: stéréovisoin et localisation par indexation d'images," Ph.D. dissertation, Université Toulouse III, January 2004.