Obstacle Detection and Tracking using Laser 2D

Danilo Habermann and Claudio Garcia
Polytechnic School of São Paulo University,
Av. Prof. Luciano Gualberto, trav. 3, nº 158, Cidade Universitária,
São Paulo/SP, Brazil - CEP 05508-900
(e-mail: danilo.habermann@gmail.com, clgarcia@lac.usp.br).

***Abstract*— An obstacle detection and tracking system using a 2D laser sensor and the Kalman filter is presented. This filter is not very efficient in case of severe disturbances in the measured position of the obstacle, as for instance, when an object being tracked is behind a barrier, thus interrupting the laser beam, making it impossible to receive the sensor information about its position. This work suggests a method to minimize this problem by using an algorithm called Corrector of Discrepancies.**

I. Introduction

Since 2002, the DARPA (Defense Advanced Research Projects Agency), encourages universities, colleges and companies to develop autonomous vehicles. One of the goals of the U.S. government is to make a third of its fleet of military vehicles autonomous by 2015 (Urmsom et al, 2008). Essential functions for these vehicles are detection and tracking of obstacles, which are helpful for the navigation of autonomous vehicles as well as interfaces to semi-autonomous vehicles, showing the driver the obstacles around the car and alerting him of possible collisions (Miranda Neto; Zampieri, 2008).

Video cameras are ideal to identify road signs and painted lines. However, they have difficulty in accurately estimating the distances of various objects that come into the picture. This problem can be solved with the use of laser sensors, which have very high precision measurements of distance and can be used regardless of luminosity conditions.

In industries, obstacle detection can be used in applications related to safety of personnel and machinery, such as the handling of containers at ports and autonomous vehicles in mines (Roberts, Cork, 2001).

Our goal is to provide a system capable of detecting and tracking obstacles in a road, using a laser sensor attached to the ground. Detection and tracking become more complex when the sensor is in motion. The reason to work with a fixed sensor is to avoid the noise resulting from the vehicle movement. Thus, the initial focus is on obtaining reliable software for detection and tracking of fixed and moving obstacles. Practical applications for this type of situation are more restricted; however, this study will be the basis for a future use of sensors mounted on vehicles in motion, mapping and tracking obstacles around them. An application of the suggested configuration is, for example, detecting and tracking people in restricted areas. The position of the intruder could be detected and tracked, enabling monitoring him with the use of a mobile camera.

Section 2 presents the characteristics of the 2D laser sensor used. In Section 3, the main components of a system for detecting and tracking obstacles are explained. The results obtained in tests with moving vehicles can be checked in Section 4. Finally, Section 5 provides the conclusions obtained with the use of the software and presents suggestions for future work to improve the system.

II. Laser Sensor

The laser sensor Sick LMS-291 used in this work is employed in several studies on autonomous and semi-autonomous vehicles. The sensor operates by measuring the return time of pulses of a laser light. A pulsed laser beam is emitted. When it reaches an object, the reflection is recorded by the receiver module of the sensor. The time between transmission and reception of the pulse is directly proportional to the distance between the sensor and the object. With the aid of an integrated rotating mirror, the laser is emitted in all directions, mapping an angle of up to 180° (Figure 1).

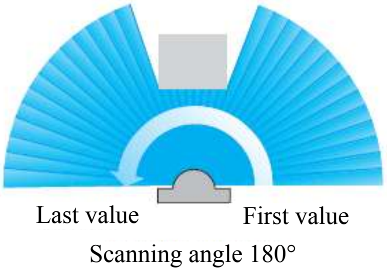


Fig. 1.Representation of 180° laser scan - Sick LMS-291.

The contour of the target is determined in response to impulses received. Measurements are taken every 0.25 °, 0.5 ° or 1 °. Its range may reach 80 m, depending on the reflectivity of the object. The measurement data are sent to a computer in real time in polar or rectangular coordinates.

This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-CAPES.

978-0-7695-4231-7/10 $26.00 © 2010 IEEE
DOI 10.1109/LARS.2010.21

III. Overview of the Obstacle Detection and Tracking System

Figure 2 presents an overview of the system architecture. The laser sensor is installed in a location where one wants to detect and track objects of interest and data is processed by the detection and tracking software, represented by the modules inside the dashed rectangle in Figure 2. The result can be observed in a man-machine interface (MMI), allowing a person to monitor the presence and movement of obstacles within the range of the sensor in real time. These data can also be sent to another system, such as for data fusion with other sensors in applications of autonomous vehicles (Cheng et al, 2007).

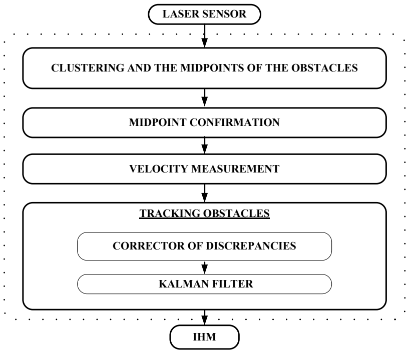


Fig. 2. System overview.

A. Clustering and the midpoints of the obstacles

The size of the obstacles is significantly influent, because when one receives the raw data coming from each sensor scan, the grouping of points to define an obstacle will be according to the distance between two consecutive points. Points belong to the same group when the distance between two neighboring points is less than a predetermined value D (Mendes et al., 2004).

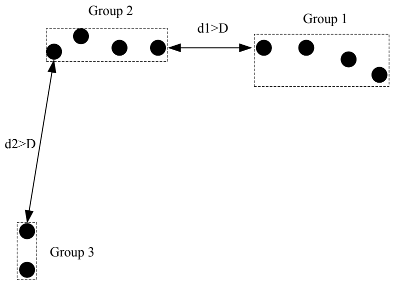


Figure 3. Representation of 3 groups or obstacles for a determined *D* value.

Analyzing Figures 3 and 4, one can verify the influence of a pre-defined distance *D* as the basis to perform the confirmation of an obstacle. Depending on the value of *D*, more or less obstacles can be detected, which does not match reality.

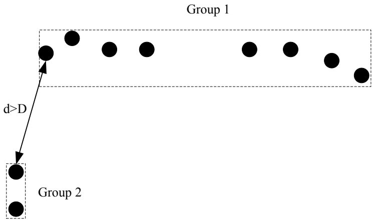


Figure 4. Representation of 2 groups or obstacles for a certain *D* value that is larger than in Figure 3. Groups 1 and 2 in Figure 3 were merged into one.

After grouping the data points in order to define obstacles, the midpoint of each of them can be found.

B. Midpoint confirmation

In each scan, several obstacles can be identified. These are numbered according to their position in order to accomplish the tracking. The object at the extreme right becomes the first obstacle, thus the object on the left is considered the last one. This is due to the operation of the sensor, which scans the area from right to left, as shown in Figure 1.

Figure 5(a) shows two vehicles approaching the sensor. Initially, only the vehicle on the left is within the range of the sensor, denoted by the dotted line. It is thus characterized as obstacle 1.

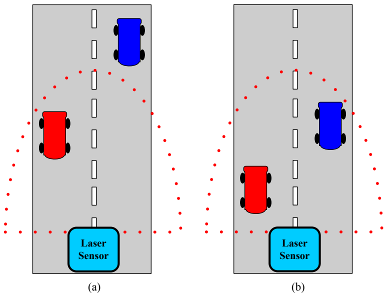


Figure 5. Two vehicles approaching the sensor.

Moments later, both vehicles are within the sensor range, as can be seen in Figure 5(b). In this situation, the vehicle on the right will be considered initially as the first obstacle, being on the right of the sensor and the other vehicle as the second. This problem would turn the tracking unfeasible, since the vehicle on the left, which was the first, becomes the second obstacle in the presence of another car. To circumvent this problem it is necessary to create a routine which confirms the midpoint of the obstacle being tracked, represented by the rectangular coordinates, xm_{ik} and ym_{ik}, in which *i* represents the obstacles in each data acquiring of the sensor and *k* represents the current scan. Consider $P_k = [p_{1k}, p_{2k}, \dots, p_{ik}]$ the vector that stores the midpoints of the *i* obstacles found in the *k*-th scan, in which $p_{ik} = (xm_{ik}, ym_{ik})$. P_{k+1} is the vector that stores the midpoints of the obstacles encountered in the (k+1)-th scan. To ensure that the obstacles being tracked always have the same position within the vector, it is necessary to compare all the elements of vectors P_k and P_{k+1}. Take the first element of P_k and calculate the distances between it and all the elements of P_{k+1}. The element of P_{k+1} that has the shortest distance, becomes the first element of an auxiliary vector *Vaux*. This is done with the other elements of P_k. In the end, it is assumed that $P_{k+1} = Vaux$. This fixes the initial problem, because the difference in position of the obstacles for the next scan is usually very small.

C. *Velocity Measurement*

The calculation of the object speed in the x and y axes is performed according to equations (1) and (2), in which h represents the time scan of the system.

$$\dot{x}m_{ik} = \frac{xm_{ik} - xm_{i(k-1)}}{h} \qquad (1)$$

$$\dot{y}m_{ik} = \frac{ym_{ik} - ym_{i(k-1)}}{h} \qquad (2)$$

This information is useful to implement the Kalman filter, presented in subsection 3.4.2.

D. *Tracking obstacles*

The tracking is primarily conducted by storing the midpoints of the obstacles encountered in vector P. However, these positions are obtained based only on data from the sensor. Noise or other disturbances such as occlusions can generate a lot of error in the system. Thus, it is necessary to apply other techniques to increase the software reliability. In order to perform the tracking, the Kalman filter and a function to correct major errors of positioning, here called the Corrector of Discrepancies or simply Corrector, is employed.

1) *Corrector of Discrepancies:* the main concern here is to track vehicles. It is known that their movements follow a certain pattern. Thus, it is possible to identify abrupt changes in behavior. When a vehicle being tracked is in an area where there is a barrier between it and the sensor, it is not possible to detect its position. In this case, its last positions have small errors because during the moments at which the vehicle is partially covered, there is error in the determination of the average position and hence its speed tends to decrease, as can be observed in Figure 6. As the Kalman filter is not able to correct very sudden changes, one needs to make an improvement before applying the filter.

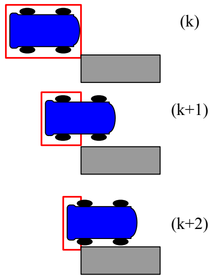


Figure 6. Vehicle passing behind a barrier.

For each obstacle, the average distances between their last five positions m_i is observed. When the distance between two successive positions of an object $p_{i(k-1)}$ and p_{ik} is greater than three times m_i, then apply the Corrector, as explained below. The average of the last ten values of the velocities of the obstacles, Vxm_i and Vym_i obtained from $\dot{x}m'_{ik}$ and $\dot{y}m'_{ik}$, that are the speeds in the x and y axes after the application of the Kalman filter, are frequently updated. Noting that there is a discrepancy between $p_{i(k-1)}$ and p_{ik}, disregard the value p_{ik} and assign to it a new value, as in equations (3) and (4).

$$xm_{ik} = xm'_i(k-1) + (Vxm_i)h \qquad (3)$$

$$ym_{ik} = ym'_{i(k-1)} + (Vym_i)h \qquad (4)$$

These new values of xm_{ik} and ym_{ik} are used in the Kalman algorithm to find xm'_{ik} and ym'_{ik}, as explained next.

2) *Kalman Filter:* it is a recursive algorithm often used to track obstacles. It is applied to estimate the state variables of systems represented by linear state equations. To use this filter, that the system is considered to be linear and disturbed by Gaussian noise.

$$x_k = A_k x_{k-1} + B_k u_k + \varepsilon \qquad (5)$$

$$y_k = C_k x_k + \delta \qquad (6)$$

Equation (5) shows how the state variables x_k evolve according to the previous state and control action. Matrix A_k describes how the state evolves from k to k +1, based only on the previous state. Matrix B_k describes how control action u_k modifies state k to k +1. These two matrices are deterministic. This equation is used to model the state in the Kalman filter. Equation (6) models the state variables and measures must also be linear and disturbed by Gaussian noise. Vectors δ and ε in equations (5) and (6) represent the process and measurement noise, respectively. They are considered Gaussian, with zero mean, independent and with covariance R and Q, respectively.

In order to analyze the noise produced by the measuring device, one thousand samples were collected from a barrier that was fifteen meters away from the laser sensor. Subtracting 15 meters from the values found, a vector with the sensor noise is obtained. According to Figure 7, the data are very close to a normal distribution. Thus, it is possible to say that the measurement errors of the sensor follows this distribution and the Kalman filter can be applied.

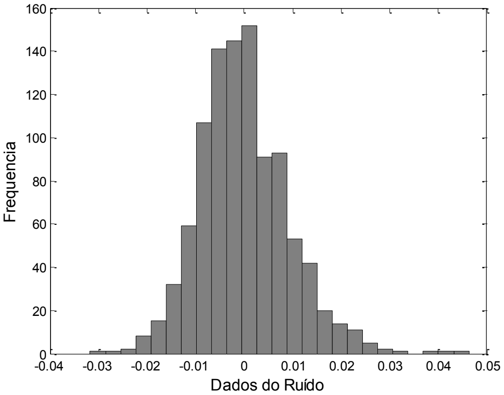


Figure 7 Histogram of the noise related to the sensor LMS-291 with an object 15 m away.

The result of the Kalman filter is the belief that an obstacle may be in a certain position and is represented as a normal distribution with mean μ_k and covariance matrix Σ_k. Below is the algorithm of the Kalman filter used in this study (Thrun, 2006).

```
Income: $\mu_{k-1}, \Sigma_{k-1}$

Prediction: $\bar{\mu}_k = A_k \mu_{k-1} + B_k u_k$
            $\bar{\Sigma}_k = A_k \Sigma_{k-1} A_k^T + R_k$

Correction: $K_k = \bar{\Sigma}_k C_k^T (C_k \bar{\Sigma}_k C_k^T + Q_k)^{-1}$
            $\mu_k = \bar{\mu}_k + K_k (y_k - C_k \bar{\mu}_k)$
            $\Sigma_k = (I - K_k C_k) \bar{\Sigma}_k$
Outcome $\mu_k, \Sigma_k$
```

The following parameters are used:

$$y_{ik} = [xm_{ik}, \dot{x}m_{ik}, ym_{ik}, \dot{y}m_{ik}]^T, \; B = 0$$

$$A = \begin{bmatrix} 1 & h & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R = \begin{bmatrix} \varepsilon & 0 & 0 & 0 \\ 0 & \varepsilon & 0 & 0 \\ 0 & 0 & \varepsilon & 0 \\ 0 & 0 & 0 & \varepsilon \end{bmatrix}, \qquad Q = \begin{bmatrix} \delta & 0 & 0 & 0 \\ 0 & \delta & 0 & 0 \\ 0 & 0 & \delta & 0 \\ 0 & 0 & 0 & \delta \end{bmatrix},$$

$$\Sigma_0 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \quad \text{and} \quad \mu_{i0} = [xm_{i0}, 0, ym_{i0}, 0]^T.$$

The result of the filter is stored in the vector $\mu_k = X'_{ik} = [xm'_{ik}, \dot{x}m'_{ik}, ym'_{ik}, \dot{y}m'_{ik}]^T$.

IV. Results

This section presents some practical tests carried out to verify the performance of the obstacle tracking software. In all the tests, the following equipment was used:

- ✓ Laser sensor Sick LMS-291, configured for a range of 80 m, scanning 180° and angular resolution of 0.5°, positioned 0.50 m above the ground.
- ✓ Vehicle Ford Fiesta 2007, silver color, dimensions: 420 x 176 x 146 cm.

During the tests, the driver only had the help of the speedometer inside the car to keep at the stipulated speed, thus generating error due to the inaccuracy of the apparatus and the difficulty in maintaining the vehicle direction and speed constant. $\delta = 0.03$ and $\varepsilon = 0.01$were used. The value of δ was extracted from Figure 7. The scan time of the sensor was 215 ms.

A Test with a vehicle at a constant speed of 20 km/h in a straight path

The first test was with a vehicle approaching the sensor with constant speed of 20 km/h. The test track had a slight curve at the beginning and 40 m in line, as shown in Figure 8. The vehicle began accelerating in order to reach the straight path at a speed of 20 km/h and remained at this rate until it passed the sensor. For the x-axis, the vehicle was 3 meters from the sensor.

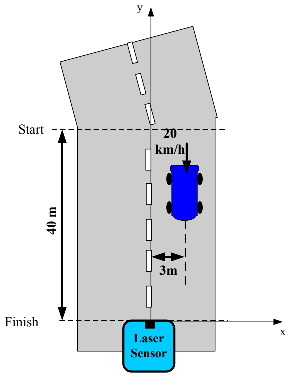


Figure 8. A vehicle approaching the sensor at 20 km/h.

Figure 9 shows the results of the tracking software. The diamonds represent the position of the vehicle obtained directly from the sensor data for 42 scans. The points are the positions of the vehicle obtained by the tracking module, employing the Corrector of Discrepancies and the Kalman filter. The first position of the vehicle was observed 51.6m away from the sensor. To prevent processing undesirable points, items that were out of the track were discarded.

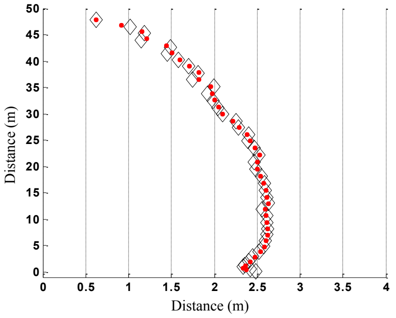


Figure 9. Estimated position of a vehicle traveling at 20 km/h.

It may be noted that even after the vehicle entered the straight path 40 meters away from the sensor, there was a position variation of about 1 m in the x-axis. This is basically due to two factors. The first one is due to human error in driving the vehicle and the second one is because the system has a small error to obtain the midpoint. Figure 10 shows the measurements made by the laser of the same vehicle at different distances. It is possible to see in Figure 10(a) that only three points are obtained from a vehicle at about 41.5 m from the sensor, and it was only possible to see the front of it. When the car is about 1.5 meters away from the sensor, many points are obtained and the outline of the front and the left part of it can be clearly seen. As in this work the vehicle position is obtained through the center of all these points in a scan, it will generate a certain uncertainty about the actual location of the object.

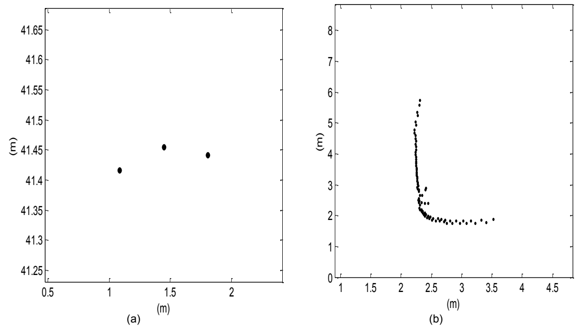


Figure 10. Measurements made by a laser sensor of the same vehicle. (a) only 3 points are obtained by laser at 41.5m and (b) several points collected at a distance of about 1.5m, the contour of the front and the left side of the vehicle.

Figure 11 shows a comparison between the midpoints obtained from the position of the vehicle in the y-axis (+), estimated by the software, using the tracking module and the position calculated by the equation governing an uniform rectilinear motion at a speed of 20 km/h (solid line). The first 28 data collected from the time the vehicle reached a distance of 40 meters from the sensor were considered. Under these conditions, the average velocity in the y-axis, obtained by the software, is 20.57 km/h and the maximum absolute error between the measured position and the line S = 40-5,55*t* was 1.48 m, with an average difference of 0.8 m.

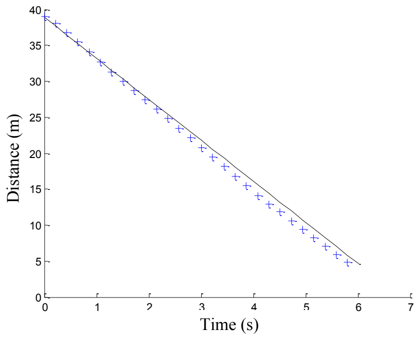


Figure 11. Comparison between measured vehicle position and equation S=40-5,55*t*.

4.2 Test with a vehicle at 20 km/h passing by a stationary vehicle

The objective of this test is to see how the system behaves when there is a barrier between the sensor and the obstacle being traced. The vehicle which is farther from the laser sensor in Figure 12 represents the vehicle to be tracked, which travels at 20 km/h in the direction of the x-axis and the other vehicle is stopped.

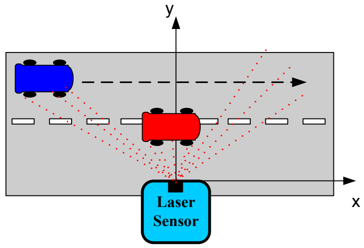


Figure 12. Sketch of the test with 2 cars, one is stationary and the other is moving.

In Figure 13, there are the midpoints of the vehicle position in the x-axis obtained by using the Kalman filter and the Corrector (+) and the line segment S =- 32.5 +5.55*t* (solid line), which represent the movement of the vehicle, considering a speed of 20 km/h. There are minor differences between the lines, which are due to the passage of the vehicle behind a barrier, hiding it from the sensor and also because of the uncertainty of the actual displacement of the vehicle.

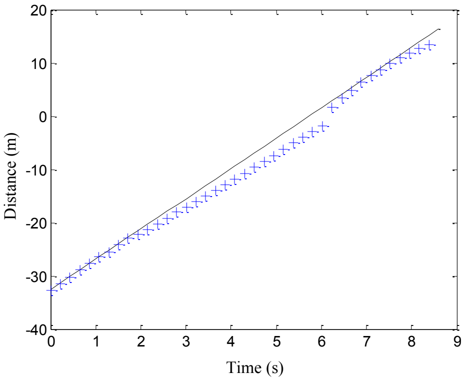


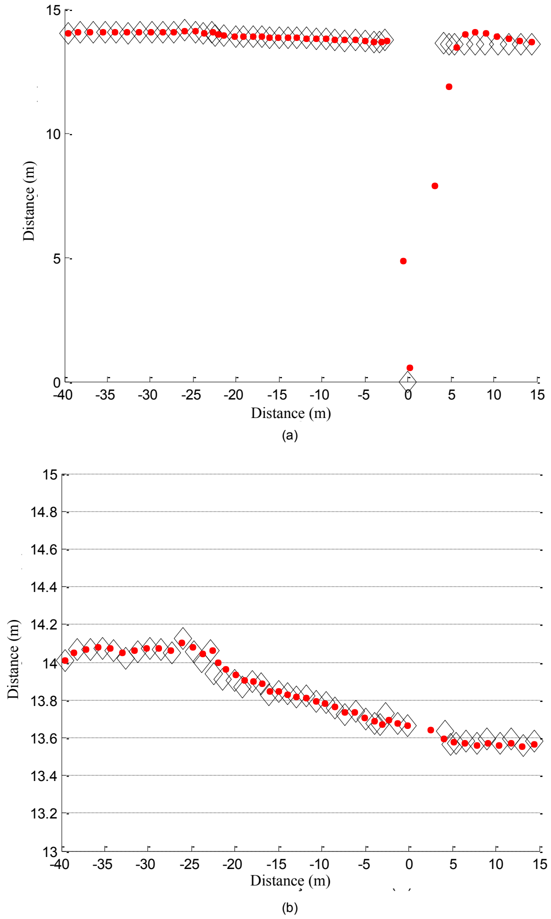
Figure 13. Comparison between the estimated vehicle movement (+) and equation S=-32,5+5,55*t* (continuous line).

In Figure 14(a), it may be noted that there is a large spacing between the diamonds in the range of -4<x<4. These diamonds represent the recorded position of the vehicle without using the tracking module. The dots represent the position of the vehicle with the use of the Kalman filter, however, without using the Corrector of Discrepancies. The laser sensor can not see the moving vehicle as it passes behind a barrier, returning two null values. The Kalman filter can not correct the measured position of the vehicle in this case. Before entering the barrier, the car moves 1.10 m on average between each scanning. The distance between the last point measured before the vehicle is placed behind the barrier and the first to point out of it was 6.8 m. The use of the Corrector and then the Kalman filter yields a better tracking in this case. The results can be seen in Figure 14(b). There is a decrease of the maximum distance between two successive points, which in this case is 2.6 m instead of 6.8 m.

V. Conclusion

One of the needs of autonomous and semi-autonomous vehicles is to obtain accurate and real-time information from the environment, in order to detect the presence of obstacles. Thus, many sensors are used, one of them being the laser.

It was observed that the use of the Kalman filter in conjunction with the function Corrector of Discrepancies provides good results, especially when the vehicle being tracked is passing behind a barrier. It is necessary to test this system with vehicles doing other kinds of maneuvers to verify the performance of the system.



(a)

(b)

Figure 14. Vehicle position at each scanning (a): only with the Kalman filter. (b): With Corrector and Kalman filter.

Acknowledgment

The authors thank CAPES, AGSP and DMC-UNICAMP.

References

[1] Cheng, H.; Zheng, N.; Zhang, N.; Qin, J. and Wetering, H. (2007) Interactive road situation analysis for driver assistance and safety warning systems: framework and algorithms. IEEE ITSS, vol. 8, n. 1, pp. 157-167.

[2] GraphPad (2008) “How useful are normality tests?”

[3] Mendes, A., Bento, L. C. and Nunes, U. (2004) Multi-target detection and tracking with a laserscanner. Intelligent Vehicles Symposium, University of Parma, Parma, Italy, June 14-17.

[4] Miranda Neto, A. and Zampieri, D. (2008) Sistema de navegação (semi)-autônomo de assistência ao condutor, Roberts, J e Corke, P. (2001) Obstacle detection for a mining vehicle using a 2D laser, CSIRO Manufacturing Science & Technology.

[5] Thrun, S.; Burgard, W. and Fox, D. (2006) Probalistic Robotics, MIT Press.

[6] Urmson, C.; Anhalt, J. and Bagnell, D. (2008) Autonomous driving in urban environments: Boss and the Urban Challenge, Journal of Field Robotics, vol. 25, n. 8, pp. 425–466.