Actas do Encontro Científico
3º Festival Nacional de Robótica - ROBOTICA2003
Lisboa, 9 de Maio de 2003.

TRACKING OF MULTI-OBSTACLES WITH LASER RANGE DATA FOR AUTONOMOUS VEHICLES

Sérgio Santos, José Eduardo Faria, Fernando Soares, Rui Araújo and Urbano Nunes

Institute of Systems and Robotics
University of Coimbra – Polo II
P-3030-290 Coimbra, Portugal
{rui, urbano}@isr.uc.pt

Abstract: Object tracking consists of estimating the trajectories and behaviour of objects, which will obviously play a role of the utmost importance in the future development of collision avoidance algorithms. This work tackles a group of modules that enlightens us on the positions and velocities of existing objects, in the visualisation area of a Laser Measurement System (LMS), using their scans. Scan processing is possible thanks to several modules (Segmentation, Static LMS and Moving LMS), which allow the tracking of several visible objects with the LMS, in a nutshell estimating their positions and velocities according to the same LMS, that can move and rotate.

Keywords: Tracking, Detection, Sensor, Kalman Filtering

1. INTRODUCTION

The matter concerning the tracking of objects and people assumes a paramount role in the understanding of the environment involving the vehicle (in order to control it) resulting in the safety of the vehicle itself, its passengers and other road users.

Research has been made in multi-objects tracking. In (Fod, *et al.*, 2002) object tracking with a static sensor is performed. To improve tracking it's useful to use cameras to determine the position and orientation of objects (Patric and Christensen, 2001). Tracking with a sensor in movement requires a new approach towards the issue stated in (Dietmayer, et al., 2001; Sparbert, *et al.*, 2001; Fuerstenberg and Willhoeft, 2001; Fuerstenberg, *et al.*, 2002).

This paper describes a tracking system constituted by the following modules: segmentation, static LMS and moving LMS. This system allows estimating the position and velocity of existing objects in the environment thanks to a number of scans provided by a laser measurement system (LMS).

2. SEGMENTATION

The goal here is to identify the limits of possible existing objects detected by the LMS and if so, to filter and provide additional information on the object in analysis. The main idea is to subdivide the readings obtained in each scan, into small sets of neighbouring points (segments), taking into account the proximity between two consecutive points of the scan.

A segment is, hence, a set of measurement values (points of the scan) close enough to each other that, due to their proximity, probably belong to the same object.

<u>Segmentation criterion</u> (based on (Dietmayer, et al., 2001)): two consecutive points, distant from the LMS, r_a and r_b, belong to the same segment as long as the distance between them fulfils the following expression:

$$|r_a - r_b| \leq C_0 + r_{\min} \cdot \frac{\sqrt{2\ \ (1-\cos(\phi)\ )}}{\operatorname{cotg}(\beta)\cdot\cos\left(\frac{\phi}{2}\right) - \operatorname{sen}\left(\frac{\phi}{2}\right)} \quad (1)$$

where r_{min} = min{ r_a, r_b } , and ϕ is the angular resolution of the LMS. Angle β and constant C_0 are parameters of segmentation sensibility refinement (tuning). β was introduced to reduce the dependence of the segmentation with respect to the distance between the LMS and the object.

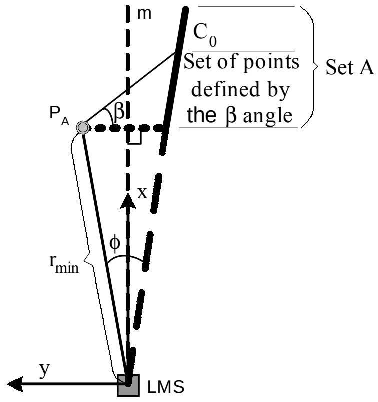

Fig. 1. Illustration of the segmentation algorithm. Set A - Set of values that the farthest point can take to belong to the same segment as the nearest point P_A.

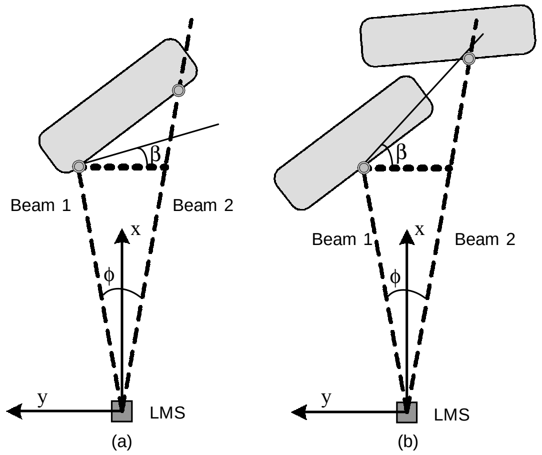

Fig. 2. Different β angles. (a) β too short – points of the object face don't belong to the same segment. (b) β too high – two points detected from different objects belong to the same segment.

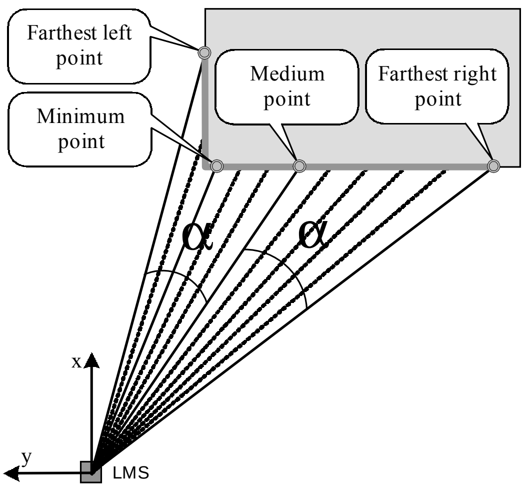

Fig. 3. Different points localization in a segment. α is the medium angle that correspond to the medium point.

If C_0=0, then β gives the maximum absolute inclination with respect to the perpendicular to the dashed line *m* (see Fig. 1), that an object's face can have to belong to a segment. *m* is the median line between the two consecutive LMS beams.

If r_{min} is short the set of points defined by the β angle is small and if it is high, the set of points defined by the β angle is high. If β is too short, some objects will not be detected (see Fig. 2-a). If β is too high, then a segment can represent more than one object (see Fig 2-b).

After the subdivision of each scan in segments, a selection of several points of the segment (see Fig. 3) and the computation of the visible dimensions of the object take place.

3. TRACKING WITH SATIC LMS

3.1 Reference point choice

In order to retrieve a reliable tracking of the object, it is vital to define a certain point which can be easily identifiable in successive scans, corresponding to the same point in the object – this is a so called reference point. Through segmentation, the information regarding several points of each segment (farthest right, farthest left, minimum and angular medium) (Fig. 3) is gathered and it must be guaranteed that the movement of the reference point is compatible with, and adequately represents, the object's movement.

The minimum point would be in principle a good candidate to represent the detected object motion behaviour, because it is always visible in the object. But two problems can arise (Fig.4):

1)-In particular situations of the object and of its faces position with respect to the LMS beams, the minimum point changes in the object face;

2)-The detected minimum point may not coincide with the real minimum point of the object, due to the existence of an angular aperture between consecutive beams of the LMS (see Fig.3).

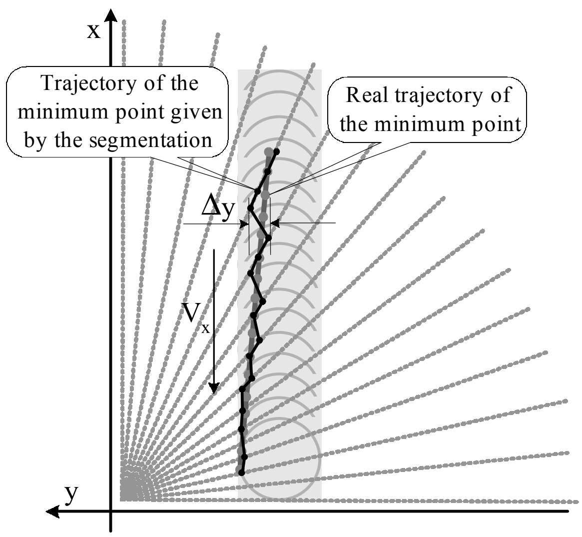

Fig. 4. Consequence of the existence of an angular aperture between beams, in the case of a circular object.

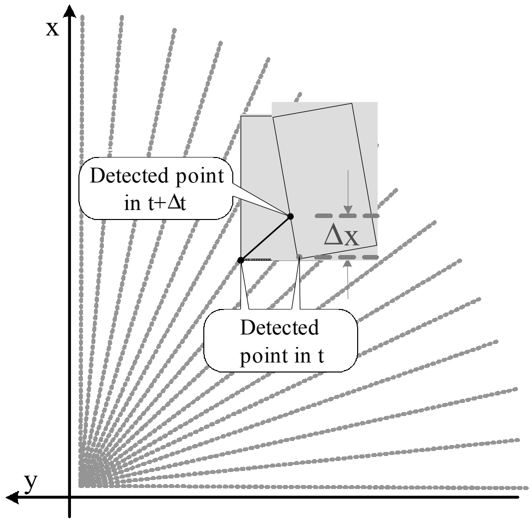

Fig. 5. Virtual displacement (Δx) of the medium point detected due to a rotational movement of the rectangular object.

The medium point corresponds to the medium angle of the detected object angular range (Fig.3). This point is the one that better behaves in conformity with the movement of the object. However this point also presents the same problems as the minimum point. In particular, if a rectangular object describes a little rotation, the detected medium point changes in the object (Fig. 5).

Taking into account the minimum and medium point characteristics, a virtual point has been defined (x coordinate of the minimum point, y coordinate of the medium point). For the example illustrated in Fig. 5, the virtual point is the one that better represents the movement of the object (see Fig. 6). This is the segment reference point used in the objects tracking.

3.2 Segment-object pair identification

It is necessary to identify the segment-object pair in an unquestionable fashion taking into account the segment reference point.

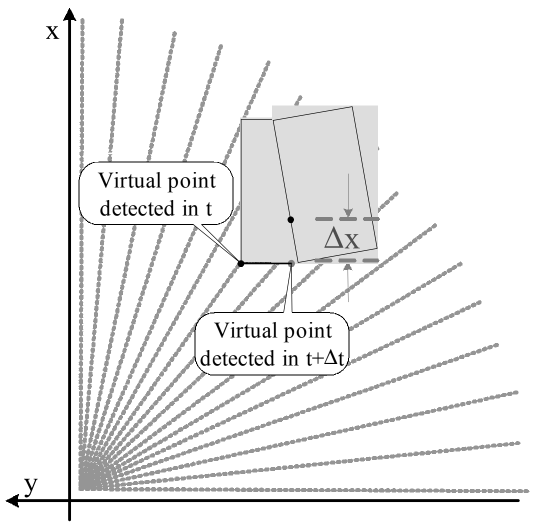

Fig. 6. Virtual point (x coordinate of the minimum point, y coordinate of the medium point).

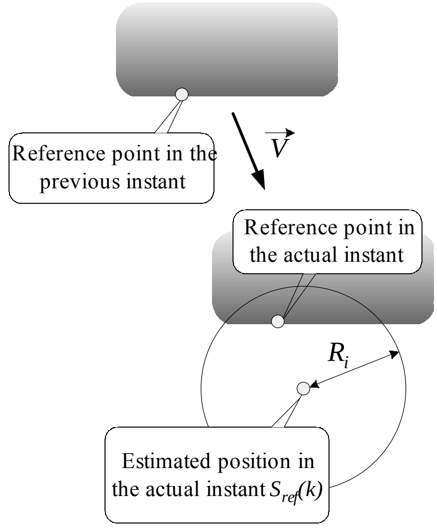

Fig. 7. Identification of the segment-object pair in polar coordinates. The interest region is a circle with a given radius R_i and centred in $S_{ref}(k)$.

The estimated position (S_{ref}) of an object reference point at the current time instant k must be obtained (for this purpose a Kalman filter is used as described in Sec. 3.3).

Now the estimated position of each object and the segments' reference points of the present scan are known. Based on the $S_{ref}(k)$ of each segment, an interest region is defined, in which the search of existing objects will be performed, searching for the match of segment $S_{ref}(k)$ (see Fig. 7). Suppose that several potential objects are found. The one with most similar dimensions (defined by the points: farthest right, angular minimum and farthest left) to the segment (object boundary detected) in consideration is chosen.

A problem arises when object reference point occlusion occur. To detect situations of object occlusion, it is determined if the object reference point will be behind of any other or out of the measurement zone of the LMS in the next scan. In such situation two options are possible: 1) object velocity update will be frozen, or 2) at least an indication of how velocity reliability is given.

3.3 Object kinematics model

This subsection describes the velocity estimate. To perform this estimate, a Kalman Filter containing a kinematics model of the object (white noise acceleration model (Kohler, 1997)) is used. The motion is considered to be the superposition of an ideal basic motion with, for example, constant velocity and white noise. The white noise illustrates the acceleration that is time varying. The discrete time state equation of this kinematics model with sampling period h ($h=t_{k+1}-t_k$) is given by (2)

$$\begin{bmatrix} s_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_k \\ v_k \end{bmatrix} + w_k$$
$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} s_k \\ v_k \end{bmatrix} + e_k \quad (2)$$

where s_{k+1} and v_{k+1} are the position and the velocity, respectively, in the instant $k+1$, w_k is the vector (considered white noise, in this model) that represents stochastic data with respect to the acceleration, y_k is the output of the model and finally, e_k is the observation error. The model described above is applied independently to the x and y coordinates, with separated Kalman Filters speeding the calculus. A special attention was drawn to the process and noise covariance matrices values for use in the Kalman filter. The filter covariance matrices are chosen similarly to (Kohler, 1997). The covariance matrix Q_k of process noise error is

$$Q_k = Q_0 = \frac{a_{\max}^2 h}{6} \begin{bmatrix} 2h^2 & 3h \\ 3h & 6 \end{bmatrix} \quad (3)$$

where a_{max} is the spectral amplitude of the white noise and Δt is the sampling period. Since the acceleration is identified as white noise, a_{max} is the maximum amplitude of the object's predictable acceleration. This matrix may be applied for any filter models with translational motion of constant velocity and random acceleration. If Q is high (Q with high values), it means that the system model is able to follow high changes. Objects with different kinematics properties are modelled using different values for a_{max}.

Since the model described by (2) is applied independently to the x and y dimensions, the two involved measurement error covariances are the following scalars: $R_x = \alpha_{e,x}^2$ and $R_y = \alpha_{e,y}^2$.

Apart from the LMS sensor range measurement error, the variances ($\alpha_{e,x}^2, \alpha_{e,y}^2$) also incorporate the error due to the angular aperture between beams (see Fig. 4).

The initial Covariance Matrix $\mathbf{P_0}$ of the estimation error is:

$$P_0 = \begin{bmatrix} \sigma_{s,s}^2 & \sigma_{s,v}^2 \\ \sigma_{s,v}^2 & \sigma_{v,v}^2 \end{bmatrix} \quad (4)$$

where $\sigma^2_{s,s}$ is the variance of the position estimation error; $\sigma^2_{s,v}$ and $\sigma^2_{v,s}$ are the cross variances of the position and velocity estimation errors; $\sigma^2_{v,v}$ is the variance of the velocity estimation error.

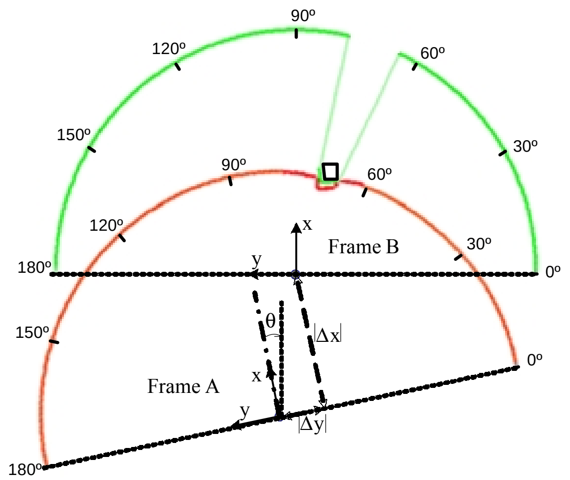

Fig. 8. Referential transformation between two reference frames which represent two different positions of the LMS.

4. TRACKING WITH A MOVING LMS

Since the LMS is to be used in an automobile vehicle that may rotate and/or translate, it is imperative to consider the movement of the LMS in the tracking of the objects. In fact when performing a Kalman filtering update cycle all the variables involved (including present and next state) must be expressed in the same reference frame. This problem can be solved using the transformation matrix that relates the present state variables (expressed in frame {B}) with the previous state variables (expressed in frame {A}), as depicted in Fig. 8.

So, if the coordinates $({}^A x, {}^A y)$ of a point in the frame {A}, are known, then, its coordinates $({}^B x, {}^B y)$ in frame {B} are given as follows:

$$\begin{aligned} {}^B x &= \left({}^A x - \Delta x\right) \cdot \cos\theta + \left({}^A y - \Delta y\right) \cdot \sin\theta \\ {}^B y &= \left({}^A y - \Delta y\right) \cdot \cos\theta - \left({}^A x - \Delta x\right) \cdot \sin\theta \end{aligned} \quad (5)$$

where vector $(\Delta x, \Delta y)$ represents the position of frame {B} with respect to, and expressed in, frame {A}. θ is the angle of rotation from frame {A} to frame {B}. The velocities will have to be projected to this new system of axes too:

$$\begin{aligned} {}^B v_x &= {}^A v_x \cdot \cos\theta + {}^A v_y \cdot \sin\theta \\ {}^B v_y &= {}^A v_y \cdot \cos\theta - {}^A v_x \cdot \sin\theta \end{aligned} \quad (6)$$

4.1. Data flow

Several modules constitute the tracking architecture each one with an associated model. The block diagram in Fig. 9 describes the relation between the several models.

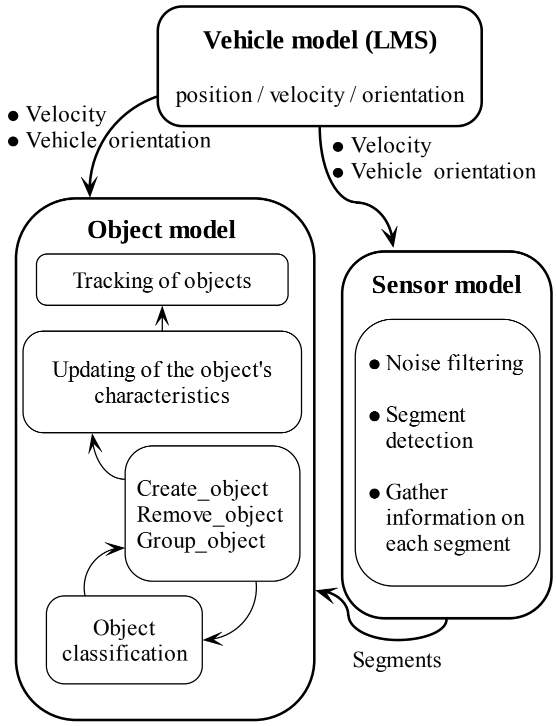

Fig. 9. System's architecture

The vehicle model is determined by its odometry in combination with the position in which the sensor is placed. The computed data is used in the other 2 models in order to carry out the tracking. The sensor model intends to identify the configurations of the sensor itself and by doing so, tuning the segmentation and filter algorithms. Finally, the object model is designed to explain the generic characteristics of the target objects (maximum velocity, maximum acceleration, dimensions) so as to refine the tracking algorithm.

5. EXPERIMENTS

Experimental results regarding two different types of environment are reported: a real experiment and a simulation using Matlab. In the real life environment the algorithm's performance was tested with a static LMS providing scans at a rate of approximately 5 Hz.

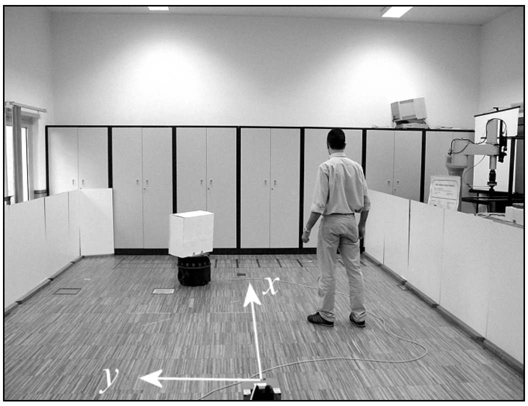

Fig. 10. Picture of the experimental environment with the scout robot carrying a white cardboard box.

The following objects were used: cardboard boxes, placards, a SCOUT robot, etc. The knowledge of all the objects' behaviour, including the LMS observation model, made possible the programming of the experiment with the moving LMS in simulation. In both experiments, the angular resolution of the LMS was configured to 0.5º and the reach to 8 m (indoor).

5.1. Real environment experiments

As an example is presented a real experimental environment, in which it is intended to track a person and a cardboard box placed on top of the Scout robot (Fig. 10). The following values have been used in the experiment:

$$\begin{aligned} &a_{\max} = 80 m/s^2 \\ &h = 0.2135 s \\ &R_x = R_y = 400 \\ &P_0 = \begin{bmatrix} 30 & 50 \\ 50 & 200 \end{bmatrix} \\ &Q_k = Q_0 = \begin{bmatrix} 20.76 & 145.86 \\ 145.86 & 1366.4 \end{bmatrix} \end{aligned} \tag{7}$$

The tracking algorithm was applied, taking into account the scans of the environment. Fig. 11 presents the velocity vectors of the two tracked objects, the identifying numbers of the objects (2 is a person and 3 is a cardboard box placed on the Scout robot) associated to the magnitude of the velocity in km/h.

Fig. 12 compares the real velocity with the estimated velocity of the cardboard box using respectively, the Scout robot's odometric data and the tracking values.

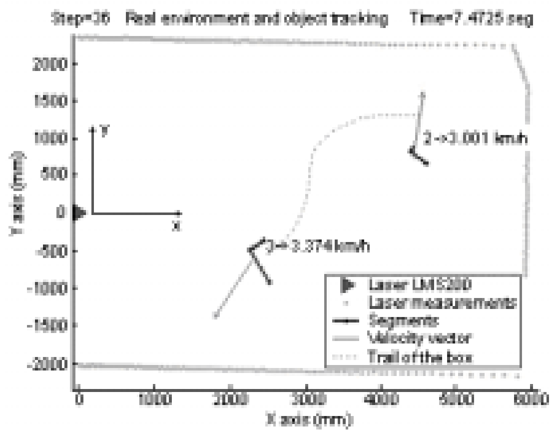

Fig. 11. Graph regarding a specific instant of the experience. We can observe that the object nº3 (cardboard box) moves at 0.937 m/s in the direction of the third quadrant; object nº2 (person) moves at 0.833 m/s in the y-axis direction. The dashed line represents the trajectory followed by the cardboard box.

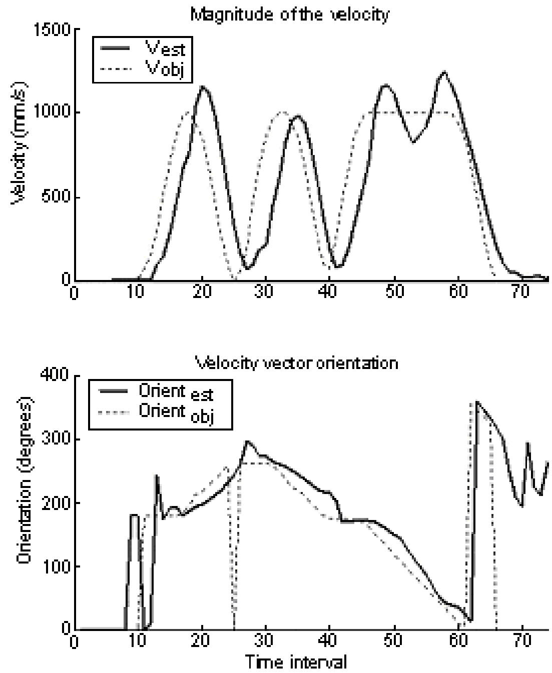

Fig. 12. Graphs that compare the absolute values of the real (V_{obj}) and estimated (V_{est}) velocities by the tracking algorithm, as well as respective orientations of the velocity vectors.

If the magnitude of the real velocity is zero, the orientation of the velocity vector is also zero. Between time intervals 47 and 57, an error occurs because of the following situation: when the object moves close to the LMS, there is a face that becomes temporarily invisible producing an oscillation on the reference point. The estimate of the velocity vector orientation does not have any meaning when the velocity is zero (or close to zero) as we can see near the time instant 10.

5.2. Simulated environment

The simulator used allows the creation of vehicles of various forms and dimension and the programming of its movements. It also includes the possibility of using a laser sensor similar to the LMS. This subsection presents a simulating experience, where the vehicle that carries the LMS (A) moves at the constant velocity of 8.333 m/s (30 km/h) from left to right and another vehicle (B) moves in the opposite direction with the same velocity. Using the scans of the LMS taken from the Simulator, the tracking algorithm was applied. On the right hand side of Fig. 13, the objects detected by the LMS, with their respective velocities are represented. Object B is the car moving in the opposite direction. Objects U and L are the upper and lower walls respectively. Using the tracking algorithm presented in this paper the velocity of vehicle B is estimated with reasonable accuracy.

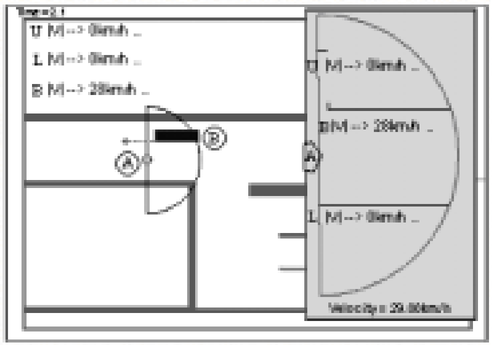

Fig. 13. Scenario with the crossing of vehicles A and B (black rectangle). The semicircle represents the area scanned by the LMS.

Fig. 14 illustrates a temporal evolution composed of time steps 1, 5, and 9 of a similar tracking experiment with the vehicle carrying the LMS moving in the opposite direction of a car (rectangle signalised as 5), both with a velocity of 30 Km/h. In step 1 the vehicle is detected for the first time. In step 5 the estimated vehicle velocity is 27 Km/h. Step 9 represents a situation where the tracked vehicle is still in the LMS field of view, and the velocity estimate is 31 Km/h.

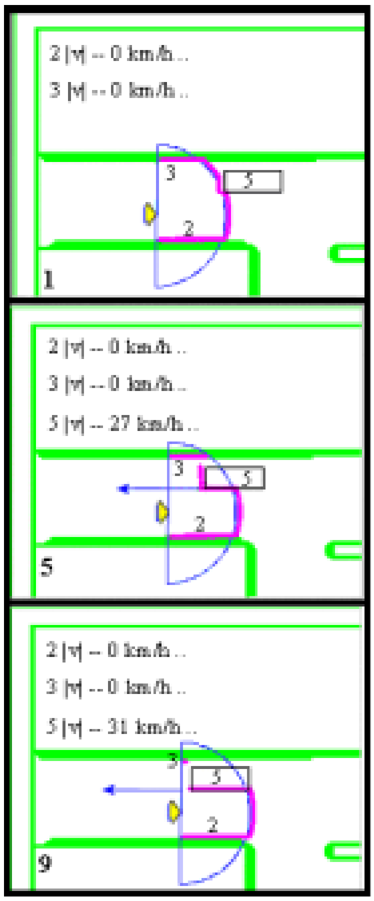

Fig . 14. A temporal evolution composed of time steps 1, 5, and 9 of a tracking experiment. Scenario with the crossing of two vehicles.

6. SUMMARY AND CONCLUSIONS

An object-tracking algorithm was developed in which the sensor itself (LMS) can be placed on a moving vehicle. It includes a kinematics model that can be adapted to the type of object detected. In order to do this, a more robust object classification is paramount. Previous knowledge on the type of the environment in addition to the increase of the rate of data acquisition are some of the developments being carried out which can improve the tracking system performance.

7. ACKNOWLEDGEMENTS

This work was partially supported by the Portuguese Science and Technology Foundation (FCT/MCES) by project LRN2002, contract POSI/SRI/42043/ /2001. In producing this project we were fortunate to receive the assistance of Alberto Vale (Institute of Systems and Robotics – IST), to whom we express our deep gratitude and appreciation for providing the LMS200 driver for Linux.

8. REFERENCIES

Dietmayer, K. C. J., J. Sparbert and D. Streller (2001). Model Based Object Classification and Object Tracking in Traffic Scenes. *IEEE Intelligent Vehicle Symposium*, IV'2001, Tokyo, Japan, pp.25-30.

Fod, A., A. Howard and M. J. Mataric (2002). A Laser-Based People Tracker. 2002 IEEE International Conference on Robotics & Automation, pp.3024-3029. IEEE, Washington.

Fuerstenberg, K. and V. Willhoeft (2001). Object Tracking and Classification using Laserscanners – Pedestrian Recognition in urban environment. *2001 IEEE Intelligent Transportation Systems Conference*, Oakland (CA), USA. pp. 453-455.

Fuerstenberg, K., K. C. J. Dietmayer and V. Willhoeft (2002). Pedestrian Recognition in Urban Traffic using a vehicle based Multilayer Laserscanner. *IEEE Intelligent Vehicle Symposium*, IV'2002 Versailles – France.

Kohler M.(1997), Using the Kalman filter to track Human Interactive Motion - Modelling and Initialization of the Kalman filter for translational Motion. Technical Report 629, Informatik VII, University of Dortmund.

Patric, J. and H.I. Christensen (2001). Pose tracking using laser scanning and minimalistic Environment Models. IEEE Transactions on robotics and Automation, Vol. 17, nº 2, pp. 138 – 147.

Sparbert, J, K Dietmayer and D. Streller (2001). Lane Detection and Street Type Classification using Laser Range Images. *2001 IEEE Intelligent Transportation Systems Conference*, Oakland (CA), USA. pp. 456-461.