Proceedings of the XII International Symposium on Dynamic Problems of Mechanics (DINAME 2007),
Varoto, P. S. and Trindade, M. A. (Editors), ABCM, Ilhabela, SP, Brazil, February 26 - March 2, 2007

The use of Obstacle Motion Tracking for Car-like Mobile Robots Collision Avoidance in Dynamic Urban Environments

Marcelo Becker [1], Richard Hall [2], Björn Jensen [3], Sascha Kolski [4], Kristijan Maček [4], and Roland Siegwart [4]

[1] PUC Minas, IPUC – Av. Dom José Gaspar, 500 – 30535-610 – Belo Horizonte, Brazil.

[2] KTH, CSC, Skolan för datavetenskap och kommunikation – SE-100 44 – Stockholm, Sweden.

[3] Singleton Technology Sàrl, Rue du Tunnel 3 – CH 1005 – Lausanne, Switzerland.

[4] ETHZ, IRIS, ASL – Tannenstrasse, 3 – CH 8092 – Zürich, Switzerland.

Abstract: One of the most challenging tasks for mobile robots is to track mobile obstacles that surround them. This task is especially difficult in outdoor environments where a great variety of obstacles may induce the robot to take erroneous decisions. The mobile robot needs as much information as possible concerning the obstacle positions and speeds (direction and magnitude) in order to plan evasive maneuvers that avoid collisions. Unfortunately, obstacles close to robot's sensors frequently cause blind zones behind them where other obstacles could be hidden. In this situation, the robot may lose vital information about these obstructed obstacles that could avoid future collisions. In order to overcome this problem an obstacle tracking module based only on 2D laser scan data was developed. Its main parts consist of obstacle detection, obstacle classification, and obstacle tracking. Different methods were evaluated for extracting data from the laser data. Geometrical feature extraction itself, i.e. lines and corners, was not found sufficient. Therefore, a motion detection module using scan matching was developed. The research was mostly conducted using a MatLab simulator that reproduces a simple 2D urban-like environment with parked and moving cars, buses, trucks, and people, buildings, streets, and trees. Aiming to adjust and validate the algorithms, some samplings of real data were carried out. The tests proved the applicability of the algorithms in real urban environments and in a near future they will be used in obstacle avoidance procedures onboard a real car-like robot.

Keywords: obstacle motion tracking, car-like robots, collision avoidance, Kalman Filter, dynamic environment.

INTRODUCTION

Making the concept of car-like mobile robots feasible is one of the biggest challenges for robotics and mechatronics researchers in the beginning of the XXI century. Since the middle of 1980s passenger cars and commercial vehicles have been considered some of the best examples of mechatronics systems. Every year onboard sensors, actuators, hardware and software that compose intelligent safety systems (also called as eSafety systems) become more popular thanks to governmental safety regulations. Unfortunately the single introduction of eSafety systems (such as anti-lock braking system - ABS, air-bags, roll stability control system - RSC, adaptive cruiser control - ACC, etc.) on board the vehicles was not enough to decrease the accident and fatality quantities in urban and freeway environments. One should keep in mind that inadequate road networks, lack of enforcement, and poorly maintained vehicle fleets also contribute for the entire safety problem. In this scenario an intelligent driving assistance system able to interfere only in peril of accidents is desired. This could be translated into a vehicle provided with various onboard sensors that are used to detect and extract environment features (e.g.: lanes, trees, guard-rails, people, other vehicles, etc.). These features are then used as input data for the vehicle controller that classifies the features and generates the evasive maneuvers, if it is needed. Therefore, a full autonomous behavior may allow the driver and passengers execute other more relevant or relaxing tasks during the travel. This would also make path planning more efficient, while increasing the safety in urban and freeway environments. Events such as the DARPA Great Challenge (2006) and ELROB (2006) and projects like SPARC (2006) are paving the way for the use of promising technologies in civil vehicles. Today it is possible to dream of freeways and urban environments interconnected by high-tech networks will allow the dissemination of fully autonomous vehicles. Traffic jams may be substituted by hundred of vehicles virtually interconnected and moving autonomously in a cooperative way. Aiming to achieve this dream scenario in near future, many robotics researchers are focusing their works on developing, transferring, and adapting techniques and approaches initially developed for indoor and outdoor mobile robots to car-like mobile robots. There are several problems to be solved in this area, but recent developments of onboard hardware and sensors are impelling the research progresses.

Systems like the Intelligent Parking Assist (IPS) technology onboard the Toyota Prius are becoming very popular among car buyers. According to an article from a specialized magazine, around 80% of the Japanese Prius customers are opting for the IPS technology (MobileMagazine, 2006). However, systems found in the market today are not autonomous ones. They are assistive systems that use sensors, e.g. ultrasound sensors and cameras, to help the driver during the maneuvers. A system able to provide a full parking maneuver procedure, without any human intervention, is desired and will shortly become commercial. Many researchers are working on car-like robot autonomous parking problem. Nevertheless, the best ratio between desired autonomous behavior and costs when selecting the onboard hardware and software necessary for acquiring, extracting, and interpreting the environment features is to be drawn. Some examples of researches developed in this field are those carried out by Chao *et al*. (2005), Khoshnejad and Demirli (2005), Yamamoto *et al*. (2005), and Chiu *et al*. (2005). Fuzzy Logic and Artificial Neural Networks are some of the approaches used by the authors to face the problem.

In addition to these researches, the works developed by Duan (2004), Lee and Chen (2004), Low and Wang (2005), Lu and Chuang (2005), Martínez-Marín (2005), Romero-Meléndez *et al*. (2005), Thompson and Kagami (2005), Kolski *et al*. (2006), and Maček *et al.* (2006) addressed a more complex problem: the path planning task in urban environments for car-like mobile robots. The ability of planning a path based, or not, on a previous knowledge of the environment and adapting the vehicle behavior to face real traffic conditions and the presence of pedestrians itself is considerably challenging. As urban city scenarios are extremely dynamic environments, the vehicle needs as much information as possible concerning the obstacle positions and speeds to plan evasive maneuvers. Obviously, the obstacle motion tracking is essential for providing the vehicle controllers with this information. This is a very difficult task when some obstacles close to robot's sensors cause blind zones behind them. In this situation, the robot may lose vital information about obstructed (hidden) obstacles that could avoid future collisions.

The present work pins down the obstacle motion-tracking problem in dynamic urban environments by applying a Kalman Filter in order to predict the obstacle motions when they are hidden. This would allow the vehicle controller to produce a proactive behavior (i.e., hidden and seen obstacles are being taken into account when maneuvering the vehicle) instead of a purely reactive one (i.e., only seen obstacles are being taken into account). Initially a brief review of the state of art on motion tracking is addressed and the multi-obstacle motion-tracking algorithm is shortly described. Next, the Smart Car and a MatLab simulator are presented. The simulator reproduces a simple 2D urban-like environment with parked and moving cars, buses, trucks, and people, buildings, streets, and trees and was used in the initial phase of the research in order to test the algorithms. Then, the results obtained while using real data are shown. Finally, the conclusion and outlook are presented in the final section.

MOTION TRACKING

Motion tracking algorithms are being used in a great quantity of different applications, from military (e.g. missile guided systems, air-space surveillance, etc.) to civil (e.g. virtual reality systems, human-machine interaction, etc.). All these applications face challenges concerning noise sensor data and data uncertainly. In order to deal with these problems the use of Kalman Filters was already proposed in the early 1960s by Kalman (1960) and Kalman and Bucy (1961). Early studies focused on single and multi-target tracking and data origin uncertainty applied on environment surveillance also proposed the use of Kalman Filters (Sittler, 1964, Sea, 1971, and Singer and Stein, 1971).

When dealing with the incorporation of uncertainty on the data origin in tracking, one should keep in mind that she is dealing with multiple-hypothesis tracking. This means that she has on her hands a combinatorial explosion of hypothesis that usually cannot be handled in real-time (Jensen, 2004). One may find a large quantity of publications in the literature related to motion tracking using a variety of sensors (vision, laser, etc.). As our test vehicle has an onboard laser sensor used for extracting the environment data (see Section about the Smart Car), we decided to focus our literature review on systems using mobile robots with onboard laser sensors.

Nevertheless, there is a lack of publications when it comes to car-like tracking applications in dynamic urban scenarios that use laser sensor data. Pradalier *et al*. (2004) developed an interesting approach using as test vehicle a bi-steerable car called CyCab. However, they did not focus their research on multi-obstacle motion tracking, but on the integration of some essential autonomy abilities into a single application (simultaneous localization and environment modeling, motion planning and motion execution amidst moderately dynamic obstacles).

Considering indoor applications, there are several works that focused the motion-tracking problem based on laser sensor data. Shulz *et al*. (2001) addressed the application of multi-obstacle motion tracking algorithms onboard mobile robots for tracking moving persons. They applied a sample-based representation of the joint probability density function (SJPDF) of all moving obstacles to avoid the combinatorial explosion of multi-hypothesis tracking. If on one hand they could combine SJPDF with a local occupancy grid and show the tracking of several persons through temporal occlusions in well structured environments, on the other hand, the use of SJPDF requires the knowledge of the tracked obstacle quantity. It means that they needed to use a Bayesian filter and a time sequence of the moving features quantity to estimate the tracked obstacle quantity. In the end, their motion tracking algorithm had to deal with local minima, non-linear relations due to the increased occlusions number, local occupancy grids, and the use of probabilistic filters to filter static objects, what could become difficult to handle in real-time for non-structured environments. Kluge *et al*. (2001) presented a strategy for analyzing the robot-human interaction scenario. It was based on a set of prototypical situations in crowed public environments and consists of scene analysis, tracking, action recognition, and intention reasoning procedures. Illmann *et al*. (2002) extended this analysis by applying local person density and tree-based vector quantization. Bennewitz *et al*. (2002-a and 2002-b) developed a system to segment tracks of person in a common household environment. In this work, the tracks were obtained from static laser sensors and the segmentation process was based on the Expectation Maximization approach. Their purpose was the detection of the intent of a person based on its path. Jensen (2004) addressed the multi-object tracking problem using single Kalman Filters for each individual problem. In order to deal with data association he used validation gates and to solve the measurement-track assignment he utilized linear programming techniques. Our approach is based on (Jensen, 2004) and extends it for urban-like environments focusing on tracking pedestrians and vehicles (e.g.: cars, trucks, and buses).

Figure 1 resumes the approach developed for urban-like environments. The laser raw data acquired by the SICK laser sensor is used as input data for the obstacle detection algorithms. Initially the data is segmented in order to extract the environment features and to detect motion. Features that are not moving are considered static and stored in a local map that is also used for detecting motion. After detecting the mobile obstacles, a process of data association is applied to start new tracks or update existing tracks. Tracks that are hidden for more than one second are deleted. Next both procedures, obstacle detection and tracking, are explained in detail.

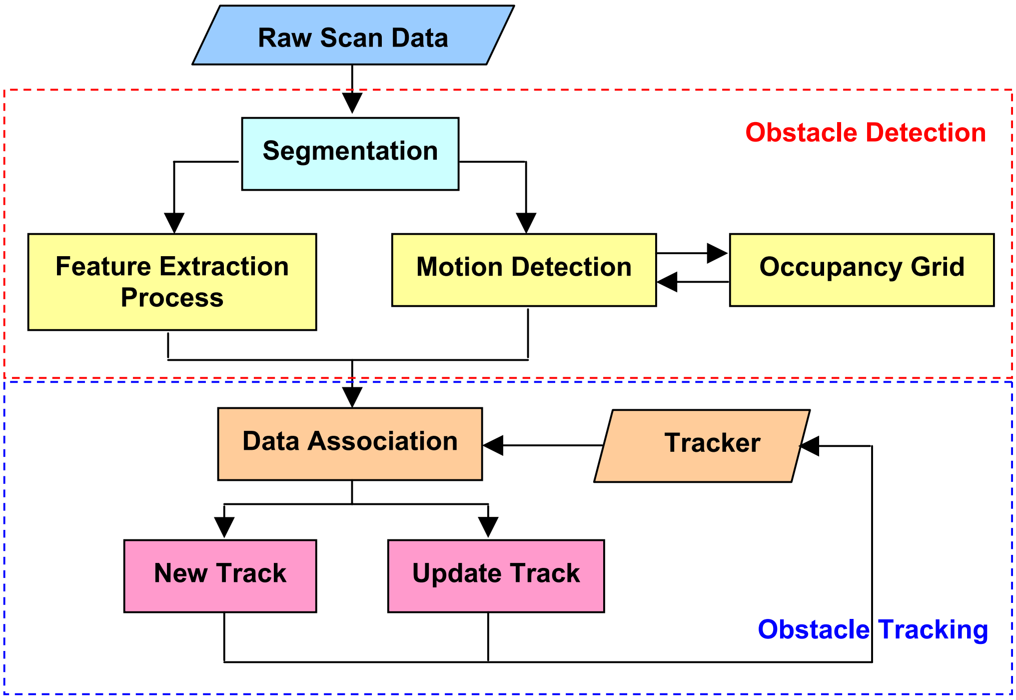


Figure 1 – Flowchart representing the General Structure of the Tracker Algorithm.

Obstacle Detection

There are several approaches in the literature for detecting objects using a laser scanner. This work focuses on detecting moving obstacles like vehicles and pedestrians. For a fixed sensor this task is straightforward since it is possible to compare two consecutive scans and immediately determine which points remains in the same spot and which do not. For a mobile sensor, like the one onboard a mobile robot, the task is somewhat more difficult due to the translation and rotation caused by its displacement. Take for example a static obstacle that is seen by only one point in the first scan. In the next scan the sensor has moved a little and the changed viewing angle results in the point being seen at another spot on the same obstacle. This phenomenon can give a false impression that the static obstacle is actually moving. In order to avoid this it is necessary to take into account, as precise as possible, the vehicle position. To perform this task, we utilized the Motion Filter approach (Lindstrom and Eklundh, 2001; Wang and Thorpe, 2002).

Segmentation

When processing data from a laser scanner it becomes necessary to group measurements that belong to the same object. The scans will consist of length measurements at equidistant angles and therefore it is very likely that two points that are close to each other also belong to the same object. Likewise two consecutive measurements that are far away are likely to imply that a change in observed object has occurred. Since it is necessary to define a distance threshold there is always the risk of making errors in the segmentation process, either by creating one single segment out of two or more close objects or by dividing an object into more than one segment. The segmentation makes it possible to do further processing on the different segments or point clusters. This normally consists of classification by size, dynamic status, or geometrical features (MacLachlan, 2004).

Motion Based Approach

Basically, the idea is to compare two scans apart by some time interval, Δt, trying to match them in order to determine which points are static and which are dynamic. By using this approach one can suppress spurious readings from the static environment, giving a better input to the tracker. The scan matching is done by storing the previous scan and comparing it to the new one. For each point in the new scan, distances to points in the old scan are calculated and compared to a threshold value (in our case, ± 10 cm).

If a match is found the point is likely to be static. One exception is when points are matched to a different area on a moving obstacle, for instance along the side of a passing vehicle. This problem was later solved by checking the dynamic classification in the first scan. If a point was labeled dynamic at that time it is likely to be dynamic in the next scan. However, if no match is found one can not say for sure that the point is moving without further processing. The point could have been occluded in the first scan or out of range due to vehicle motion between scans. Therefore we make use of the free area that is the space between the car and the obstacles or scan range. If a point seen in the second scan is not found but within this area in the first scan, thereby observable, it is labeled as moving. A segment containing a certain amount of points labeled as moving can be sent to the tracker, after its center point is calculated.

During the test phase using the MatLab Simulator (see Simulator section for its details) we observed that the simple use of scan matching wasn't sufficient for dealing with spurious readings. One reason for this behavior was the translation and rotation movements of the sensor. Aiming to increase the detection of static points a local map was created. At first the map was implemented as a position vector, storing in each scan the coordinates of all points labeled as static ones. New scans were then checked against the map and when a match was found, the weight for that map point was increased. In order to keep down the size of the map each point kept track of its own age. When a terminal age was reached, e.g. after 10 seconds, the point would be deleted from the map. The static map helped to improve the motion detection but some disadvantages were noticed. The representation of the map as points was one of them. Measurements are never exact and therefore it is difficult to determine how many map points are needed to represent an obstacle. If we add all new static points we get a very large map which slows down the matching process but if we only add points that aren't matched to the map before we will certainly miss useful information. A method that turned out to be a better approach to the Local Mapping was the Occupancy Grid (Elfes, 1989). Thanks to the change to the use of an Occupancy Grid we could improve and simplify the scan matching process. As the occupancy grid was not only used to map the static environment, but also to represent present and future predictions of the tracker output, it was necessary to classify and model the obstacles that were being tracked.

Obstacle Classification

By classification, we refer to the process of determining to which one of the two basic classes a moving segment belongs: vehicles or pedestrians. While working with the MatLab Simulator (see Simulator section for its details), pedestrians were rarely detected by more than two or three scan data. Since the sensor is placed about 50 cm from the ground it usually detects one point on each leg. Should the pedestrian stand close to the sensor it is easier to distinguish a contour but this was rarely the case. An obvious difference between vehicles and pedestrians is their size so we decided to base the classification on it. We used the standard deviation of each segment to represent its size:

$$\mu_X = E(X) = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \mu_Y = E(Y) = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{1}$$

$$\sigma_X^2 = \frac{1}{n}\sum_{i=1}^{n} (\mu_X - x_i)^2 \qquad \sigma_Y^2 = \frac{1}{n}\sum_{i=1}^{n} (\mu_Y - y_i)^2 \tag{2}$$

Where: X and Y are stochastic variables, x and y are measured points on a segment, n is the number of points and μ and σ are expected values and standard deviations respectively. Checking the norm of the standard deviations against a predefined threshold yields the classification into vehicle or pedestrian:

$$\sigma_{norm} = \sqrt{\sigma_X^2 + \sigma_Y^2} \tag{3}$$

Obstacle Tracking

A robot navigating in an environment with other moving obstacles needs some kind of information concerning its environment in order to avoid collisions. Some systems just prohibit movement in directions, which will bring the robot too close to an obstacle regardless of how the obstacle moves (Siegwart and Nourbakhsh, 2004). For efficient path planning it is, however, much better to know more about the dynamic state of the obstacles. The way to achieve this goal is by tracking them and estimating their future positions (states). By tracking it is possible to measure the dynamic state of an obstacle, i.e. its position and velocity, and with this information predictions can be made on the obstacles future positions (Bar-Shalom and Fortmann, 1998; Blanc *et al.*, 2005).

In an urban environment, the laser measurements are subject to noise, which we have to suppress by filtering. It is also usual to expect obstacles being hidden by other obstacles. However, it is possible to continue the track of these obstacles until they are seen again and thereby minimizes the risk of the path planner allowing a collision. Each obstacle in this setup is characterized by its center of gravity, or at least the center of gravity for the part that is being seen. Since the scanner provides a 2D output this point is described by (x, y) coordinates. The x and y velocities were also introduced in the state of the dynamic object for tracking the motion of each obstacle and predicting its continued path. The state vector then becomes:

$$\bar{v}_{state} = (x \quad \dot{x} \quad y \quad \dot{y})^T \tag{4}$$

But, from now on we will simply denote it as *x*. The measurements, containing only the x and y values, are labeled *z*.

Kalman Filter

The Kalman filter is a wide spread technique for estimating the state of a dynamic system observed through noisy measurements. The filter is recursive, which means that in every step it uses the output of the previous step for making a new prediction. It consists basically of two steps, the prediction step when estimation is made based on the old state and an update step when that estimation is updated with a new measurement. The state and measurement predictions are denoted by $\hat{x}$ and $\hat{z}$. In order to predict a dynamic system a dynamic model is used. In this study the linear constant-velocity model (Jensen, 2004), was applied. Due to its linearity it is simple to implement. Other models were considered and especially for vehicles the constant angular velocity model could improve the tracking for vehicles making turns. However, such implementation would require the more complex Extended Kalman Filter, EKF. It is also beneficial to keep a simple, not too specialized, model since the tracker is dealing with obstacles with different dynamic properties (e.g.: pedestrian and vehicles). More details concerning the implementation of Kalman Filters may be found in (Kalman and Bucy, 1961; Bar-Shalom and Fortmann, 1998; Jensen, 2004).

Data Association

Each track must be associated with new measurements when available if the filter is working properly. There are many approaches for data association that are designed to suit different scenarios. A Joint Probability method (Bar-Shalom and Fortmann, 1998) for example, is an interesting choice when sampling data in a cluttered environment because it takes into account the fact that the closest sample isn't always the right one in a multi-target scenario. The intuitive approach in this case was the Nearest-Neighbor Standard Filter, NNSF (Bar-Shalom and Fortmann, 1998). The NNSF algorithm iteratively calculates all the distances between tracks and measurements. The distance is represented by the Mahalanobis distance, MD:

$$m = vS^{-1}v^T \tag{5}$$

MD is the same as the Euclidian distance if the covariance matrix is the identity matrix, *I*. Each track is then associated with the closest measurement if it does not exceed a predefined threshold. However, problems may occur if the application deals with unambiguous association. An optimization can be carried out by using integer programming. In this case, MD is collected in a matrix where one tries to minimize the total sum provided there can be only one value selected from each column and each row (Jensen, 2004).

Path Prediction

In order to include future possible positions for tracked obstacles in the grid it was necessary to calculate possible paths for the obstacles and move them along those paths to simulate their whereabouts for some step in time (in this setup, 1 s). As mentioned earlier, dynamic modeling for people motion is not always trivial (Jensen, 2004) but since pedestrians are likely to travel at rather low speeds, the constant-velocity model was considered sufficient for predicting their paths. On the other hand, for vehicles the following approach was applied: most vehicles use only their front wheels to turn. The steering kinematics can be modeled using the Ackermann Geometry (Gillespie, 1992), but we make use of a simpler model. The angle of the wheels relative to their straight ahead position is called the steering angle, henceforth noted θ. Due to mechanical reasons most vehicles have a maximum steering angle that is far less than 90^o. When maintaining constant steering angle, $\theta \neq 0$, the vehicle is moving on a circular track. A simple car model (LaValle, 2006), gives us the Eq. (6):

$$\tan\theta = L/R \tag{6}$$

Where: *L* is the distance between the front and rear wheel axles and *R* is the radius of the circular path with center in Instantaneous Center of Motion - ICM. Continuing with deriving the angular velocity in the path we have:

$$d = \frac{\Delta\phi}{2\pi}.2\pi.r = \Delta\phi.r \tag{7}$$

Where: *d* is the distance along the circular path, same as $v.\Delta t$, and $\Delta\phi$ is the change in angular orientation ($\Delta\phi = \omega.\Delta t$ is the angular velocity). Further simplifications:

$$v.\Delta t = r.\omega.\Delta t \quad \rightarrow \quad \omega = v/r \tag{8}$$

Combining Eqs. (6) and (8) yields:

$$\omega = \frac{v}{L} \tan\theta \tag{9}$$

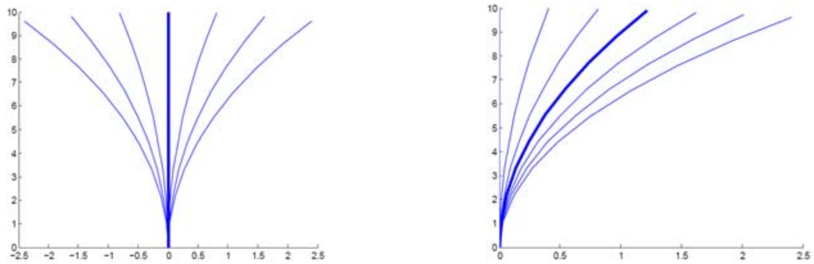
Equation (9) means that it is possible to calculate the angular velocity of an obstacle knowing its velocity, steering angle and axle distance. Using the velocity and angular velocity the predicted path can easily be calculated. Of course the placement of axles varies between different types of vehicles but it stays in the same vicinity when considering most cars. In order to simplify the model a maximum steering angle for vehicles was set to 0.42 radians. Another constraint that was applied to limit the model was the fact that steep steering angle turns are unlikely at high speed. This is related to centripetal acceleration because, extending it to the extreme, the car would slip taking a steep curve with too high speed. In a city-driving situation it is fairly unlikely that a driver would experience more than 1 g in lateral acceleration. The equation for centripetal acceleration is:

$$a_c = \frac{v^2}{r} \tag{10}$$

Combining Eqs. (8) and (10):

$$\omega_{\max} = \frac{a_{c_{\max}}}{v} \tag{11}$$

The maximum angular velocity is now given as the least value from equations (9) and (11). Choosing a number of values in the interval ($\omega_{\max}$, $\omega_{\min}$) and then calculating predicted paths for each one gives us a set of possible locations. One limitation in the tracker is that it does not provide angular velocity for the tracks. In order to do that accurately, an Extended Kalman Filter is needed. For this reason we decided to use predicted velocity changes of each tracked obstacle in order to estimate the obstacle angular velocities. Hence, the predicted tracks are centered on the estimated turning rate, see Fig 2. The highest probability value is assigned to the middle track, $\omega_{estimated}$, then decreasing as $|\omega|$ increases.



(a) **(b)**

Figure 2 – Predicted Paths for tracked obstacles. In (a) the track is currently going along the y-axis showing 7 possible paths, ranging from left to right maximum steering angles. In (b) we estimated the current steering angle, which lets us center the predicted paths on it.

TEST VEHICLE

The Smart Car used at ASL (Fig. 3) is based on an ordinary *smart fortwo coupé* passenger car. The lab has interfaced the vehicle controller area network (CAN) so that its proprioceptive sensors, like wheel and steering encoders, can be read. The Smart Car is also equipped with GPS and a 6 degree of freedom Inertia Measurement Unit (IMU), allowing the relative movement of car being measured.

Figure 3 – Modified Smart Car used at ASL. Observe the LMS 291 SICK laser sensor assembled on its front bumper.

The IMU measures lateral acceleration in all three dimensions, angular rates up to 100°/s with a resolution of 0.025°, and lateral acceleration up to 2g with a resolution of 0.01 m/s^2. Actuators controlling steering, breaking and thrust have been installed together with exteroceptive sensors as a monocular camera and a laser range finder (SICK Laser Measurement Sensor - LMS 291), both looking forward. The laser sensor was mounted on the front bumper of the Smart Car and it has an 180° field of view with an angular resolution from 1° to 0.25°. Its maximum range is 80 meters and it has a statistical error of 10 mm.

SIMULATOR

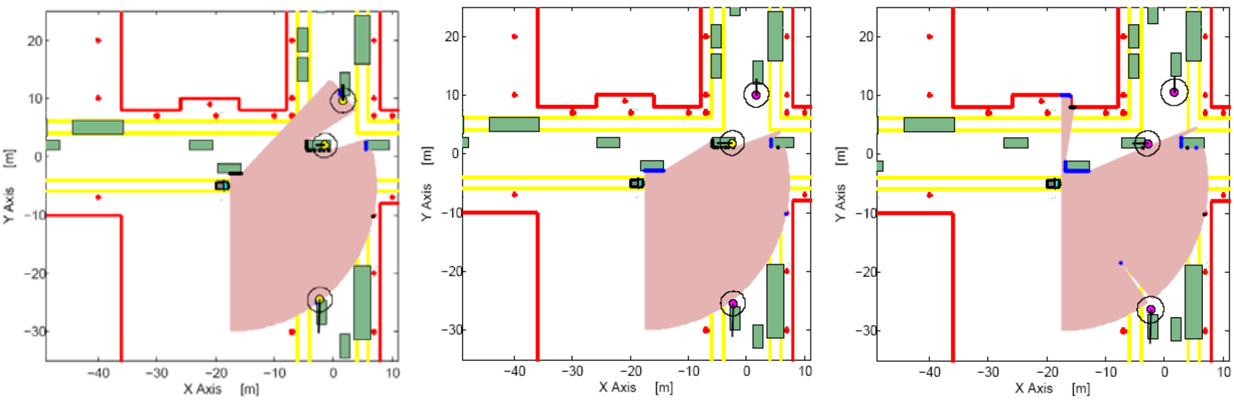
The Smart Car simulator was developed in MatLab. It reproduces a simple 2D urban-like environment (approximately 800 m x 800 m) with parked and moving cars, buses, trucks, and people, buildings, walls, streets, and trees. When using the simulator one may reproduce the 2D kinematical behavior of a modified Smart Car. The Smart Car vehicle was kinematically modeled by applying the Ackerman steering geometry (Gillespie, 1992) and the real vehicle dimensions. While modeling the sensors, their real characteristics were taken into account. Lines and/or arcs represent all environment static and dynamic features. The sensor data are extracted from the environment based on its geometrical description and used as input data for testing the algorithms.

Basically, the simulator uses the global position of the Smart Car in the environment for selecting a feature-window that contains all lines and/or arcs close to the vehicle. Then it simulates the laser sensor data by verifying the intersections between the simulated laser beam and the environment features, afterwards a noise signal is add to the sensor raw data vector. As the simulator was designed for testing different approaches before installing the codes in the real car, it allows the user to select and set different strategies, set points, and threshold values. Always in the beginning of the simulation a global path planner uses a graph representation of the environment and the Dijkstra algorithm (Dijkstra, 1959) to calculate the shortest global path. The simulator uses the right-hand traffic convention to produce global paths. This global path planner algorithm returns a sequence of intermediate positions from the vehicle initial position to its goal position. After initialized, the simulation can be stopped any time by clicking down the mouse on the pause check box, and the user can verify the path tracker controller inputs and outputs.

RESULTS

Smart Car Simulator

The research was initially developed using the Smart Car Simulator. During this phase it was possible to test different techniques for tracking the obstacles. Figure 4 shows some simulation results. The seen tracks are represented by yellow circles and the occluded or out of range tracks, by pink circles. The black lines represent the obstacle predicted positions for 1 second.



(a) (b) (c)

Figure 4 – Simulation of the Smart Car parked in an urban-like environment tracking vehicles on a street cross. Three different time steps are presented. One may observe that vehicles are tracked even if outside the sensor visible area (represented by salmon color). In this case, a pink circle represents the tracker.

All occluded tracks were kept for 1 second. Meanwhile, if they were observed again, their states were updated and they were reclassified as seen tracks. On the other hand, if they were not observed anymore after 1 second, they were deleted.

Real Data

Aiming to test and refine our approach real sample data were acquired on streets at EPFL (École Polytechnique Fédérale de Lausanne, Switzerland). After these test it was possible to refine the tracking algorithms. The sites for sampling data were chosen near parking lots at EPFL for catching as many cars and persons as possible. The sampling was also done in the late afternoon when there were people heading home by car.

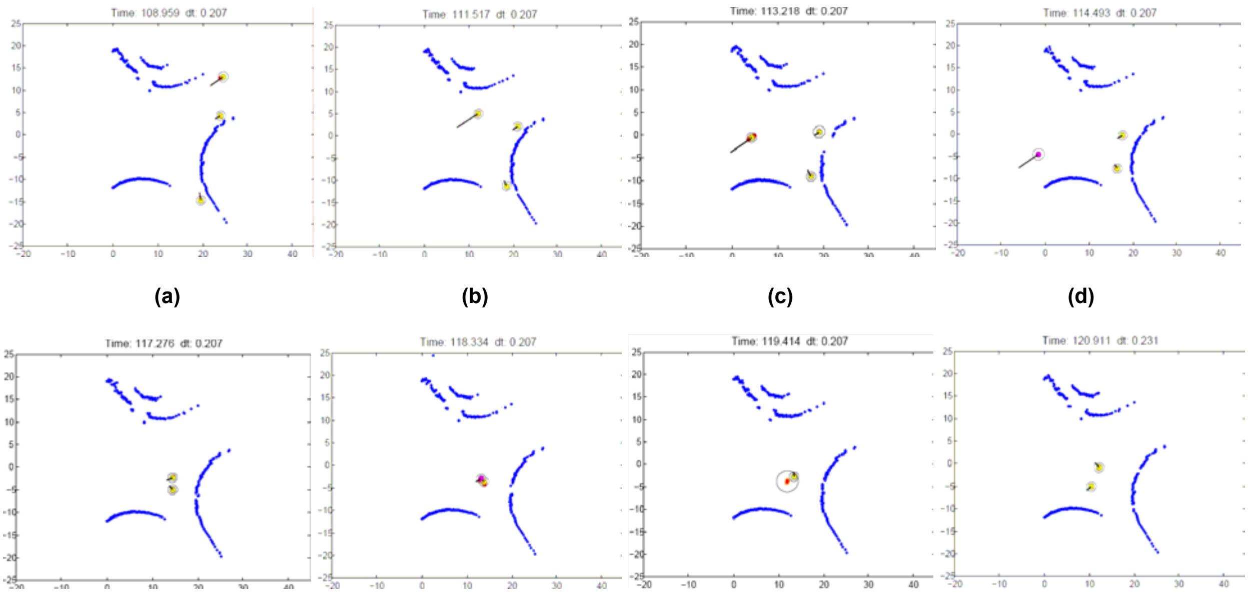
Real Data Sample

The sensor platform was placed in a corner of an intersection where the exit of a parking lot could be seen in the left image of Fig. 6. We experienced traffic and the obstacles were often passing by close to the sensor. Because of the intersection we were also lucky to capture a few cases when one obstacle occluded another. An interesting condition in this case was that we had many passing bikes. They were sometimes classified as cars, sometimes as pedestrians depending on the viewing angle. Their tire spokes also produced an interesting problem: some of the laser readings acquired by the sensor were behind the bike (the sensor laser beam passed trough the spokes and detected other environment feature that was behind the bike). Due to this, the segmentation algorithm produced erroneous input data for the feature extraction and motion detection algorithms. This problem was solved latter by refining the segmentation algorithm.

Table 1 – Real data sample

Location	EPFL South of CM complex facing northwest
Scan rate	5 scans/s
Resolution	2 measures / degree
Length	602 s
Observed Dynamic Obstacles	33

Figure 5 – Experiment location in EPFL aerial view and Sensor Field of View.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6 – Real data sample on streets at EPFL.

The scanner was placed in the origin (0,0), looking along the positive x-axis. Frames were chosen by their contents and not by direct succession. In the scenario presented, two persons were moving towards each other, one from the right and one from the bottom of the screen. A bike was also traversing, coming from the upper right corner. The tracks continued from (a) to (c) until the bike went out of range in (d), although the track is kept for one second after disappearance. The persons continued approaching until they were perceived as a single segment, (f). In (g), they could be separated again and both tracks were updated.

CONCLUSIONS

The development of a car-like mobile robot able to deal with real traffic conditions is today a challenging task. Nevertheless, thanks to teamwork of professionals from several areas, the development and adaptation of software techniques to urban dynamic environments, and the development and miniaturization of hardware and new sensors, can become feasible in a near future. Our work presented some interesting results on obstacle tracking task in dynamic urban-like environments using simulation tools and real data.

First of all we present a short review of the motion tracking techniques found in literature and highlighted the lack of publications when it comes to car-like tracking applications in dynamic urban scenarios using laser data. Then our approach and the test vehicle used, a modified Smart Car, were briefly described. The approach focused the detection, classification, and tracking tasks of vehicles (e.g.: cars, buses, etc.) and pedestrians. Initially we used the Smart Car Simulator, a MatLab-based simulation tool that reproduces a simple 2D urban-like environment, to develop the research, Then, real data sample were used in order to refine the algorithms. Finally, the results were presented.

Many hours of evaluating different methods in the Smart Car Simulator indicated that obstacle detection and tracking with a 2D laser scanner is far from a trivial task. It can be done efficiently, though many considerations to special cases must be taken into account. The algorithms make use of several parameters that have to be tuned in to the environment where the vehicle is supposed to operate. Motion detection gave good results but it is also a subject to parameter adjustment. Fluctuating measurements, such as from bushes and leaves, can cause spurious readings but the use of a static map increases the overall performance. Obstacle tracking with a simple Kalman Filter worked satisfactorily. The filter parameters also needed to be tuned in. The tracker works regardless the fact that it sees a car or a pedestrian, but for the task of path predicting, it could be useful to predict turning even for bikes, or running pedestrians.

Based on the results obtained, we initiated a new phase of the research: the integration of the tracking procedure with the obstacle avoidance procedure aiming to run both linked in real-time onboard the test vehicle.

ACKNOWLEDGMENTS

Marcelo Becker wants to thank CAPES for the financial support during his stay at EPFL, Switzerland.

REFERENCES

Bar-Shalom, Y. and Fortmann, T. E., 1998, “Tracking and data association”, Academic Press Inc., London, UK.

Bennewity, M., Burgard, W., and Thrun, S., 2002-a, “Learning motion patterns of persons for mobile service robots”, Proc. of IEEE Int. Conf. Robotics and Automation (ICRA 2002), vol. 4, pp. 3601 - 3606.

Bennewity, M., Burgard, W., and Thrun, S., 2002-b, “Using EM to learn motion behaviors of person with mobile robots”, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2002), vol. 1, pp. 502 - 507.

Blanc, C., Trassoudaine, L., and Gallice, J., 2005, “EKF and particle filter track to- track fusion: a quantitative comparison from radar/lidar obstacle tracks”, Proc. of 8th Int. Conf. on Information Fusion, vol. 2, pp. 25-28.

Chao, C-H, Ho, C-H, Lin, S-H, and Li, T-H S., 2005, “Omni-Directional Vision Based Parallel-Parking Control Design for Car-like Mobile Robot”, Proc. of the 2005 IEEE Int. Conference on Mechatronics, pp. 562 – 567.

Chiu, C-S, Lian, K-Y, and Liu, P., 2005, “Fuzzy Gain Scheduling for Parallel Parking a Car-Like Robot”, in IEEE Trans. on Control Systems Tech., Vol. 13, No. 6, November 2005, pp. 1084 - 1092.

DARPA Great Challenge, 2006, http://www.darpa.mil/grandchallenge/index.asp, accessed in May, 17, 2006.

Duan, Q. J., Wang, R.X, Feng, H.S., and Wang, L.G., 2004, “An Immunity Algorithm for Path Planning of the Autonomous Mobile Robot”, Proc. of INMIC 2004, pp. 346 – 350.

ELROB, 2006, http://www.elrob2006.org/, accessed in May, 17, 2006.

Dijkstra, E. W., 1959, A note on two problems in connexion with graphs. In: Numerische Mathematik. 1, pp. 269–271.

Elfes, A., 1989, “Using Occupancy Grids for Mobile Robot Perception and Navigation”, in: Computer, vol. 22, no. 6, IEEE Computer Society Press, pp. 46-57.

Gillespie, T.D., 1992, “Fundamentals of Vehicle Dynamics”, SAE International, ISBN: 1-56091-199-9.

Illmann, J., Kluge, B., and Prassler, E., 2002, “Statistical recognition of motion patterns”, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2002), vol. 2, pp. 1259 - 1264.

Jensen, B., 2004, “Motion Tracking for Human-robot Interaction”, Ph.D. Thesis No. 3156(2004), EPFL, Lausanne, Switzerland, p. 167.

Kalman, R.E., 1960, “A new approach to linear filtering and prediction problems”, in: Trans. of ASME, J. Basic Engineering, Vol. 82, pp. 34 – 45.

Kalman, R.E. and Bucy, R., 1961, “New results in linear filtering and prediction theory”, in: Trans. of ASME, J. Basic Engineering, Vol. 83, pp. 95 – 108.

Khoshnejad M. and Demirli K., 2005, “Autonomous Parallel Parking of a Car-Like Mobile Robot by a Neuro-Fuzzy Behavior-Based Controller”, Proc. of NAFIPS 2005 - 2005 Annual Meeting of the North American Fuzzy Information Processing Society, pp. 814 – 819.

Kluge, B., Illmann, J., and Prassler, E., 2001, "Situation assessment in crowded public environments", Proc. of Int. Conf. on Field and Service Robotics.

Kolski, S., Ferguson, D., Bellino M., and Siegwart, R., 2006, "Autonomous Driving in Structured and Unstructured Environments", Proc. of IEEE Intelligent Vehicles Symposium.

LaValle, S. M., 2006, "Planning Algorithms", Cambridge University Press, Available online: http://planning.cs.uiuc.edu/

Lee, Z., and Chen, X., 2004, "Path Planning Approach based on Probabilistic Roadmap for Sensor Based Car-Like Robot in Unknown Environments", Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics, pp. 2907 – 2912.

Li, Z. and Ghosh, B.K., 2000, "Geometric feature based 2½D map building and planning with laser, sonar and tactile sensors", Proc. of 2000 IEEE/RSJ Int. Conf on Intelligent Robots and Systems (IROS 2000), vol. 1, pp. 115 – 120.

Lindstrom, M. and Eklundh, J.O., 2001, "Detecting and tracking moving objects from mobile platform using a laser range scanner", Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2001), vol. 3, pp. 1364 - 1369.

Low, C-B and Wang, D., 2005, "Robust Path Following of Car-like WMR in the Presence of Skidding Effects", Proc. of 2004 IEEE International Conference on Systems, Man and Cybernetics, pp. 864 – 869.

Lu, H-C and Chuang, C-Y, 2005, "The Implementation of Fuzzy-Based Path Planning for Car-Like Mobile Robot", Proc. of the 2005 International Conference on MEMS, NANO and Smart Systems (ICMENS'05), pp. 467 – 472.

Maček, K., Becker, M., and Siegwart, R., 2006, "Motion Planning for Car-like Vehicles in Dynamic Urban Scenarios", Proc. of IROS 2006 - IEEE / RSJ International Conference on Intelligent Robots and Systems, October 2006, Beijing, China.

MacLachlan, R., 2004, "Tracking Moving Objects From a Moving Vehicle Using a Laser Scanner", Report, Carnegie Mellon University, Available online: http://www.cs.cmu.edu/~ram/resume/datmo_report.pdf

Martínez-Marín, T., 2005, "Learning Optimal Motion Planning for Car-like Vehicles", Proc. of the 2005 International Conference on Computational Intelligence for Modeling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05), pp. 601 – 612.

MobileMagazine, 2006, http://www.mobilemag.com/content/100/354/C6610/, accessed in June 2006.

Pradalier, C., Hermosillo, J., Koike, C., Braillon, C., Bessiere, P. and Laugier, C., 2004, "An autonomous car-like robot navigating safely among pedestrians", Proc. of 2004 IEEE Int. Conf. on Robotics and Automation (ICRA 2004) , Vol. 2, pp. 1945 – 1950.

Romero-Meléndez, C., Monroy-Pérez, F., and Vázquez-González, B., 2005, "Complexity and Path Planning for a car-like robot" Proc. of 2nd International Conference on Electrical and Electronics Engineering (ICEEE) and XI Conference on Electrical Engineering (CIE 2005), pp. 463 – 466.

Schulz, D., Burgard, W., Fox, D., and , Cremers, A.B., 2001, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association", Proc. of the 2001 IEEE Int. Conf. on Robotics and Automation (ICRA 2001), vol. 2, pp. 1665 – 1670.

Sea, R.G., 1971, "An efficient suboptimal decision procedure for associating sensor data with stored tracks in real-time surveillance systems", Proc. of 10^{th} IEEE Conf. on Decision and Control, pp. 33 – 37.

Siegwart R. and Nourbakhsh, I., 2004, "Introduction to Autonomous Mobile Robots", The MIT Press, Cambridge, Massachusetts - USA.

Singer, R.A. and Stein, J.J., 1971, "An optimal tracking filter for processing sensor data of imprecisely determined origin in surveillance systems", Proc. of 10^{th} IEEE Conf. on Decision and Control, pp. 171 – 175.

Sittler, R.W., 1964, "An optimal data association problem in surveillance theory", in: IEEE Trans. on Military Electronics, MIL-8, pp. 125 – 139.

SPARC, 2006, http://www.sparc-eu.net/, accessed in May, 17, 2006.

SICK Product Database, 2006, Available online: http://www.sick.com/

Thompson, S. and Kagami, S., 2005, "Continuous Curvature Trajectory Generation with Obstacle Avoidance for Car-Like Robots", Proc. of the 2005 International Conference on Computational Intelligence for Modeling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05), pp. 863 – 870.

Wang, C-C. and Thorpe, C., 2002, "Simultaneous localization and mapping with detection and tracking of moving objects", Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA 2002), vol. 3, pp. 2918 - 2924.

Yamamoto, M., Hayashi, Y., and Mohri, A., 2005, "Garage Parking Planning and Control of Car-Like Robot Using a Real Time Optimization Method", Proc. of the 6th IEEE International Symposium on Assembly and Task Planning: from Nano to Macro Assembly and Manufacturing - (ISATP 2005), pp. 248 – 253.

RESPONSIBILITY NOTICE