Universidade de Aveiro Departamento de Engenharia Mecânica
2010

Jorge Manuel Soares de Almeida

Seguimento de objectos dinâmicos com oclusão usando dados laser

Target tracking using laser range finder with occlusion

Universidade de Aveiro Departamento de Engenharia Mecânica
2010

Jorge Manuel Soares de Almeida

Seguimento de objectos dinâmicos com oclusão usando dados laser

Target tracking using laser range finder with occlusion

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob a orientação científica do Dr. Vitor Santos, Professor Associado do Departamento de Engenharia Mecânica da Universidade de Aveiro

Obrigado por tudo Beta

o júri

Presidente

Prof. Doutor Robertt Angelo Fontes Valente
professor auxiliar da Universidade de Aveiro

Prof. Doutor Urbano José Carreira Nunes
professor catedrático da Universidade de Coimbra

Prof. Doutor Vitor Manuel Ferreira dos Santos
professor associado da Universidade de Aveiro

Agradecimentos

Em primeiro lugar gostaria de agradecer ao Professor Doutor Vítor Santos pela orientação e pela constante motivação.

Queria também prestar um especial agradecimento ao Miguel Oliveira e ao Doutor Ricardo Pascoal pela disponibilidade, críticas e sugestões que me ajudaram em muito a melhorar o meu trabalho.

Queria agradecer aos meus colegas de mestrado Bruno Andrade, Luís Rodrigues e Luís Pereira e a toda a equipa do Atlas em especial ao David Gameiro e ao Procópio Stein pelo companheirismo e boa disposição.

palavras-chave

Seguimento de múltiplos alvos, Oclusão, Sensor Laser 2D, Filtro de Kalman Adaptativo, Regras Heurísticas e Modelos de Movimento.

Resumo

Este trabalho apresenta uma técnica para a detecção e seguimento de múltiplos alvos móveis usando um sensor de distâncias laser em situações de forte oclusão. O processo inicia-se com a aplicação de filtros temporais aos dados em bruto de modo a eliminar o ruído do sensor seguindo-se de uma segmentação em várias fases com o objectivo de contornar o problema da oclusão. Os segmentos obtidos representam objectos presentes no ambiente. Para cada segmento um ponto representativo da sua posição no mundo é calculado, este ponto é definido de modo a ser relativamente invariante à rotação e mudança de forma do objecto. Para fazer o seguimento de alvos uma lista de objectos a seguir é mantida, todos os objectos visíveis são associados a objectos desta lista usando técnicas de procura baseadas na previsão do movimento dos objectos. Uma zona de procura de forma elíptica é definida para cada objecto da lista sendo nesta zona que se dará a associação. A previsão do movimento é feita com base em dois modelos de movimento, um de velocidade constante e um de aceleração constante e com aplicação de filtros de Kalman. O algoritmo foi testado em diversas condições reais e mostrou-se robusto e eficaz no seguimento de pessoas mesmo em situações de extensa oclusão.

Keywords Multi target tracking, Occlusion, 2D Laser Rangefinder, Adaptive Kalman filter, Heuristic rules and motion models.

Abstract In this work a technique for the detection and tracking of multiple moving targets in situations of strong occlusion using a laser rangefinder is presented. The process starts by the application of temporal filters to the raw data in order to remove noise followed by a multi phase segmentation with the goal of overcoming occlusions. The resulting segments represent objects in the environment. For each segment a representative point is defined; this point is calculated to better represent the object while keeping some invariance to rotation and shape changes. In order to perform the tracking, a list of objects to follow is maintained; all visible objects are associated with objects from this list using search techniques based on the predicted motion of objects. A search zone shaped as an ellipse is defined for each object; it is in this zone that the association is preformed. The motion prediction is based in two motion models, one with constant velocity and the other with constant acceleration and in the application of Kalman filters. The algorithm was tested in diverse real conditions and shown to be robust and effective in the tracking of people even in situations of long occlusions.

Contents

List of figures

Nomenclature

KF	Kalman filter
AKF	Adaptive Kalman filter
MHT	Multi hypothesis tracking
PDAF	Probabilistic data association filter
IMM	Interacting multiple models
LRF	Laser range finder
LIDAR	Light detection and ranging
CV	Constant velocity
CA	Constant acceleration
SLAM	Simultaneous localization and mapping

Chapter 1

Introduction

Multi target tracking problem is a major issue in many applications. In indoors, target tracking can be useful in security applications like personal access control, or optimization problems like people motion flow in a mall (Zhao and Shibasaki, 2005). Another application for both indoors and outdoors situations is to assist mobile robots; by using tracking, object velocities can be extracted to then be used to implement advanced path planning and collision avoidance algorithms in mobile robots. Besides the implementation of robot path planning, this information can be used to assist human driven vehicles. Knowledge of other objects behavior and speed is paramount to the safety of any mobile agent and the world surrounding it. This is evident if one thinks of the agent has being a car. In order to drive safely in a common road the driver must be able to rapidly assess other cars motion, pedestrian motion and the motion of everything else present in the road. This information is used to determine if the driver can enter a road or pass over a crosswalk, but it is also used to avoid dangerous situations that could result in a crash. As so, one could think of a safety mechanism that refrains human commands if those commands put the vehicle at risk; this is just one simple example of the use of knowledge about the other moving agents among many more such simple applications that could greatly improve road safety. Outside roads, vehicles like electric wheelchairs could also be made autonomous or semi-autonomous, these vehicles would face problems that would require precise knowledge of other mobile agents like people or animals or even other autonomous vehicles.

In this work, an algorithm capable of providing the position and velocity of all objects in the vicinity of the sensor is presented. To accomplish the tracking a laser range sensor is used; this type of sensor presents some advantages over more common sensors like cameras but also have some disadvantages. The algorithm proposed is intended to be used in indoors situations as well as outdoors. The use of a laser sensor brings several issues that must be overcome to a successful tracking. In a multi target tracking

application, targets are free to enter or leave the workspace, so the application must be able to take into account the constantly varying number of targets. Not all new measurements correspond to objects, as some of them belong to clutter; the laser sensor can also fail to detect a target especially near the laser maximum range. With this kind of sensor, only a 2D planar cross section of the targets is visible; therefore the target shape, size and pose may change very rapidly over time. Measurement error is also a significant problem with laser data; this issue is once again more relevant near the laser maximum range. One of the biggest problems with any line of sight sensor is occlusion; in this the laser range finder is no different. In a dynamic environment objects often occlude each other; per example, if the application must track persons in an outdoors environment, objects like building columns, trees, trash cans and many others, create large occlusion areas, any person moving in the sensor range may enter these occlusion areas. Although there are some occlusions that are impossible to overcome, temporary occlusions are not and must be overcome for a reliable and robust tracking. All these factors limit greatly the techniques to exploit, and make it difficult to identify objects with high certainty. The main aim of this work is to develop an algorithm capable of perform robust multi target tracking mitigating the problems of occlusion.

The developed algorithm is based on a solution that uses a motion prediction based data association mechanism augmented with heuristic rules and uses linear Adaptive Kalman Filter (AKF) to overcome temporary occlusions.

This work is inserted into the ATLAS project (Oliveira et al., 2009) of the Department of Mechanical Engineering of the University of Aveiro. The ATLAS project started with the aim of participating in autonomous mobile robots competitions but has of late evolved into real road vehicles with the goal of developing new Advanced Driver's Assistance Systems (ADAS). The work developed here is therefore intended to be incorporated into these systems, in order to provide accurate perception of movements in the vicinity of the ego-car.

Some of the work developed here was already submitted and accept for publication in the Controlo'2010 9th Portuguese Conference in Automatic Control under the title Laser-based Tracking of Mutually Occluding Dynamic Objects.

Laser rangefinder

Today there are many laser brands and options available in the field of robotics, such as, indoors, outdoors, or dedicated to a particular system. The laser used in this work is a Hokuyo UTM-30LX, which is a general purpose laser suited for use in robotics applications. It presents a maximum range of 30 m and an angular resolution of 0.25° operating at a scan rate of 40 Hz, the laser has a detection angle of 270°.

This kind of sensor offer some advantages over more common vision systems. When performing tacking with vision systems two main problems occur, determining the position of the object and segmentation of objects. The first problem can be overcome using stereo vision or assuming a particular object size but these techniques are unreliable; in contrast, determining the position of an object is trivial using laser data. Segmenting objects from their background is also a big problem in vision systems but in laser systems due to their high accuracy this task is simpler (but not trivial, as objects that are close together may be hard to separate). But unlike a vision system the laser data contains little information: range, angle and in some cases reflectance; with these limitations much of the techniques that are employed in vision cannot be used with laser data.

1.1 Laser rangefinder intrinsic operation

A laser rangefinder (also denominated LIDAR) is an active optical position measurement sensor. The device emits laser pulses that are reflected by objects in the environment, the reflected pulses are detected by the device; the time delay between transmission of a pulse and detection of the reflected signal is used to determine the range to that object, this method is based on the Time Of Flight (TOF) principal (there are other techniques used to detect the range to the target, but TOF is the most common). In a scanning laser rangefinder the mechanical motion of a scanning mirror directs sequential measurement pulses in different directions, creating a 2D representation of the surrounding environment, an additional rotating axis will allow for a complete 3D scan. Some laser sensors are capable of performing multiple detections for each pulse; using this technique one can detect a foreground object with a faint reflection and a background object with a strong reflection. This technique is useful to detect and overcome the detection of rain or in topographical application to detect the ground vegetation; in these applications the LRF can detect both the ground floor and

the vegetation such as trees or bushes. In robotic applications with direct contact with persons, the laser beam cannot be harmful to the human beings (Class 1, eye-safe); as such the laser presents a high divergence and low power. In other applications the laser can be of a higher class and as such present higher detection range.

1.2 Problems

When performing tracking, the LRF data presents several problems that must be attended. As with all line of sight sensors the LRF is sensitive to occlusion, for instance, if a small objects moves in front of a larger background object; this occlusion will cause a false movement of the background object boundary. Another problem occurs when there's a discontinuity in range; on the transition from the nearer object to the farther one some points appear with intermediate ranges. This causes error in the detection of the object end points and may cause data segmentation problems as the two objects may be grouped together.

The LRF is also sensible to weak returns as some objects reflect poorly the laser light. For instance, black cars are typically invisible to the scanner and can only be detected at very close ranges. High angles of incidence of the laser beam with environment objects also present a problem as the return light is very small and at very high angles the laser light maybe completely reflected away from the sensor making the object invisible (Fig. 1).

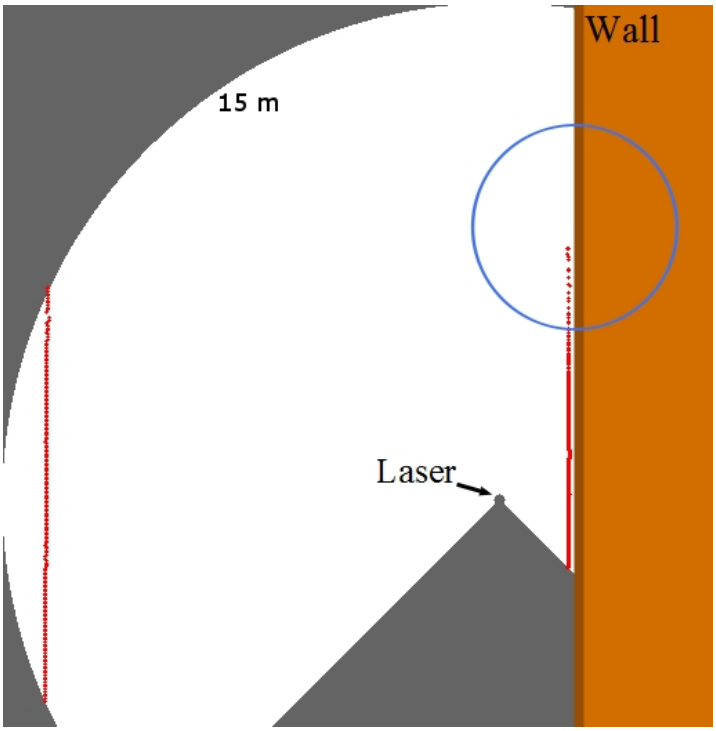

Fig. 1. The Laser stops detecting the wall in the blue marked zone.

1.3 Other applications for LRF

In the field of robotics the LRF are commonly also used to perform SLAM (Wang et al., 2007) and can also be used to detect and classify objects (mainly pedestrians or cars) (Oliveira et al.; Premebida et al., 2009). As mentioned previously, LRF are used in topographic reconstruction using airborne lasers (Fig. 2), one typical use is the shoreline reconstruction (Stockdonf et al., 2002) or tree canopy analysis for forest fire research and resource management (Andersen et al., 2005). The kind of lasers used in these applications have a much longer range than those used in robotic applications but the laser class is also much higher (not eye-safe).

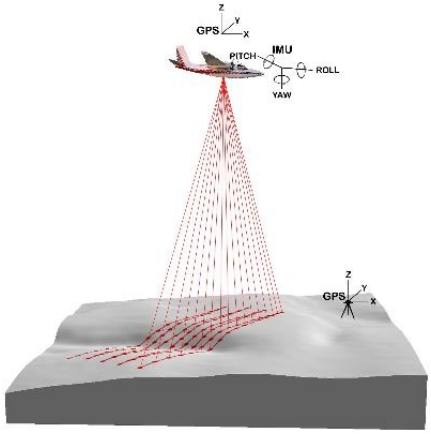

Fig. 2. Airborne LIDAR system allows the mapping of a large ground area (Reutebuch et al., 2005).

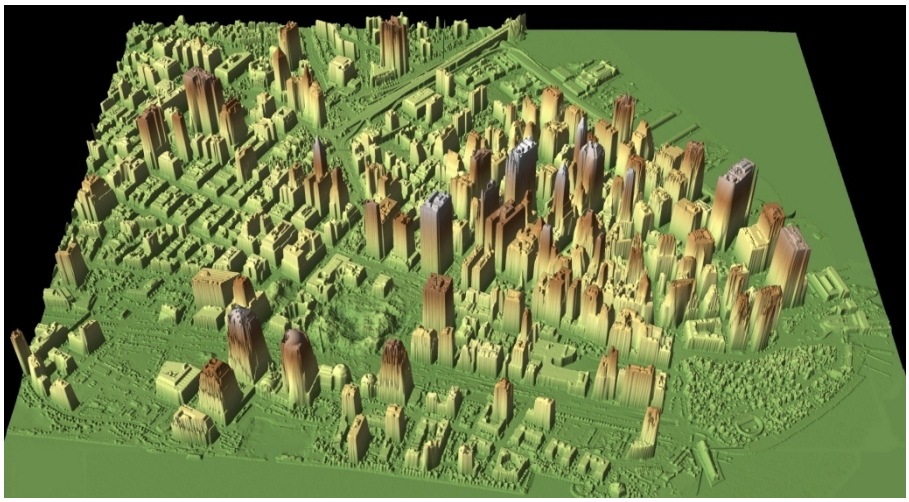

Fig. 3. 3D reconstruction of New York City in 27 September 2001 rendered by the US Army. (National Oceanic and Atmospheric Administration, U.S. Department of Commerce)

3D reconstruction of objects and environments is an important topic in many areas such virtual museums, game and entertainment industry, architecture, virtual reality, archaeology, etc. (Dias et al., 2006). In these applications the LRF is a primary tool, due to its dense and accurate measurements. This technology is also used in sports; for example, many professional golf players employ linear LRF to measure distance to holes (Fig. 4), these LRF are more limited than 2D or 3D equipments but are also much more affordable. Some of the LRF available can also supply the velocity of targets; these sensors can be used by law enforcement agents to detect law infringements by road vehicles (Samuels et al., 1992).

Fig. 4. Hand laser range finder used by Golf players (Bushnell).

Chapter 2

State of the art

Much work has been developed on the subject of multi target tracking over the past few decades (Blackman and Popoli, 1999). There's and extensive number of different approaches to tackle this problem, only the ones that occur more frequently in literature will be referenced here. One typical way of performing multi target tracking is to use cameras and computer vision. This approach has been extensively studied and many different techniques have been developed, these techniques will not be exploited in this work since we use a laser range finder; a survey on vision based object tracking is found in (Yilmaz et al., 2006).

Laser tracking has been employed mainly in two different applications: pedestrian tracking with static sensors and road vehicle tracking from a moving platform. The typical implementation contains the following main steps: data association, motion estimation and prediction. To perform data association, several different methodologies exist, and are normally based in some kind of object reconstruction from measurements. A typical data association algorithm is the Multi Hypothesis Tracking (MHT) (Reid, 1978); in this technique multiple associations are propagated over time instead of just the most likely. This allows for a recombination of previous associations based on some criteria. This algorithm is not restricted to data association, as some authors use it to propagate object classification hypothesis or object morphology hypothesis. The main issue with this algorithm is the exponential growth of number of hypothesis; in order to make this algorithm feasible, pruning and gating techniques are employed. These techniques transform the initial optimal algorithm into a sub-optimal one. Typical motion estimation and prediction is Kalman-based, but particle filters are also employed with success. Various motion models are used, constant velocity or constant acceleration being the most common, but not the only ones. Some authors also apply Interaction Multiple Models (IMM). In this technique several models are used to

represent the same system and propagated in time using the same measurements. Their combined information is used to more accurately track objects.

In (Fod et al., 2002) a people tracking approach is presented. The authors use multiple laser range finders placed at waist height in an indoors environment; their algorithm is based on simple gradient clustering methods and a nearest neighbor approach to perform data association. They model the environment into a background and a foreground model in order to obtain measurements belonging to moving objects; blobs (clusters) are created from foreground measurements using a simple gradient threshold. Their data association method has two distinct phases, a first blob association and a second object association. Low level association using nearest neighbor is preformed on the blobs from two consecutive scans, while objects characterize higher level objects constituted by multiple blobs. A set of heuristic rules are used to associate blobs to objects and deal with temporary occlusions. Kalman filter is used to smooth sensor noise and predict motion of occluded objects. Simple clustering techniques have some relevant shortcomings that will be discussed later, using nearest neighbor data association techniques also present some intrinsic problems since the nearest neighbor is not always the best association due to occlusions. Their algorithm is not able of coping with objects that present higher acceleration than a person.

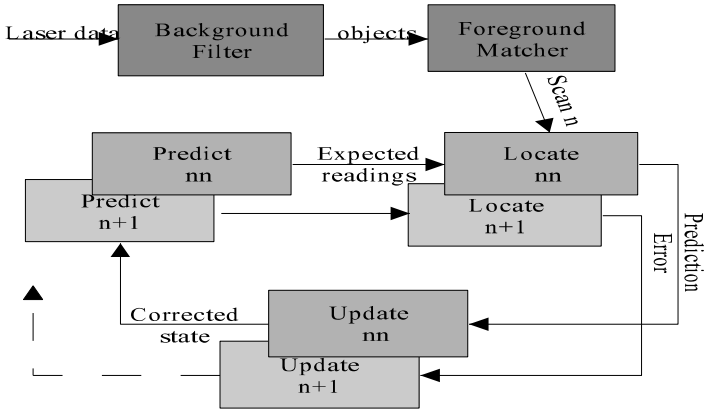

Fig. 5. Information flow in (Fod et al., 2002). A background filter is used to remove background readings; foreground blobs are matched with blobs from the previous scan; then for each blob the Kalman filter is updated and a prediction is made.

(Zhao and Shibasaki, 2005) also make use of several laser sensors to track multiple objects in a densely populated wide open environment. The laser sensors are placed at ground level and they assume that the only moving objects are people's feet. The authors argue that placing the sensor at ground level is better that at waist level due to smaller occlusions. A motion model derived for feet motion is used. Prediction is

Kalman based, using simple gating to define missed associations. A map of the physical surroundings is used to filter out static objects and is periodically updated. The authors noted that their algorithm was not robust enough to track each individual track; also, their work has some limitations since only moving feet are accounted for. Additional work by the same research group in the same topic may be found in (Cui et al., 2007) and (Cui et al., 2008).

Moving vehicles are another application that requires multi target tracking. These vehicles can be small human scale robots or full scale autonomous cars. (Arras et al., 2008) presents a technique to perform people tracking using a variation of MHT that incorporates adaptive occlusion probabilities. Estimation and prediction is Kalman based using constant velocity models. In their experiments the only moving objects were legs.

In (Kobilarov et al., 2006) the authors propose two methods for tracking and following persons in an outdoors unstructured dynamic environment. They use a laser range finder in conjunction with an omnidirectional camera mounted on a two-wheel dynamically balancing Segway Robot Mobile Platform to track and follow persons. Their first method uses the camera as the main sensor and the laser to estimate the range to the detected object; in the second approach the laser is the main sensor and the camera is the secondary. In the second method they use the laser to extract the 3d relative position of objects that may correspond to people and use camera information that corresponds to the laser object to provide additional measurements. Simple clustering techniques were employed to extract objects from laser scans; data association is performed using a probabilistic data association filter (PDAF); they employ two motion models, a constant velocity model with white noise acceleration and a nonlinear coordinate turn model with a state vector containing the target turn rate in addition to the constant velocity model state components; they noted that the second model had better performance under occlusion. Two levels of gating were applied, a maneuver gate based on the maximum possible velocity and acceleration of a moving person, and an elliptical covariance-based gate. They manage to successfully follow a person in a public park with temporary occlusions as long as the occlusion time was small.

(Petrovskaya and Thrun, 2009) propose a technique to detect and track moving vehicles. They model both dynamic and geometric properties of tracked vehicles and employ a Bayes Particle filter; by using geometric models of the tracked vehicles they avoid simple clustering techniques and some of the problems associated with them like partial

occlusion of vehicles (Fig. 6). Their algorithm was applied in their Junior robot that won the second price in the DARPA Urban Grand Challenge. Their system was successful in the detection and tracking of moving vehicles even at high speeds. They used several laser range finders including a Velodyne laser to perform the tracking; this setup is very expensive. Moreover, they do not take into account other road users like motorcycles, bicycles or even pedestrians.

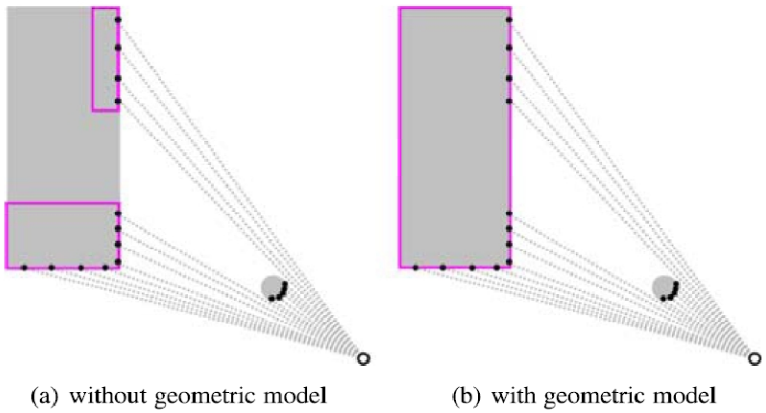

Fig. 6. The figure demonstrates the use of geometric models in order to avoid partial occlusions. Purple rectangles group together points that have been associated together. In (b) the purple rectangle also denotes the geometric vehicle model. Gray areas are objects. Gray dotted lines represent laser rays. Black dots denote laser data points. (Petrovskaya and Thrun, 2009)

In the German University of Ulm several important studies on the topic of multi target tracking have been produced over the years. Their main focus has been the application of tracking to traffic scenes in the form of driver assistance systems. In (Dietmayer et al., 2001) the authors propose a algorithm that implements a model based object classification and tracking; this algorithm incorporates a sensor model, dynamic model of the car, a street model and a model for each individual object in vicinity of the car. They implement simple clustering methods based on the distance between two consecutive measurements. To classify objects they test their dimensions against a database of known vehicles; their classification is used to set up the initial parameters for the motion models; linear Kalman filter is used to perform the tracking. A similar approach is used in (Fuerstenberg et al., 2002) but in this work a multi layer LRF is used. They also employ an algorithm that is able to avoid the disintegration of objects due to partial occlusions. In (Streller et al., 2002) the problem of egomotion is tackled using coordinate transformations. Objects are modeled similarly to the previous work and extended Kalman filter is used to predict objects position and define gating zones. In (Streller and Dietmayer, 2004a) and (Streller and Dietmayer, 2004b) a multi hypothesis approach is proposed. In this work they use simple clustering to segment

objects, but due to the problems inherent to this kind of approach they generate multiple association hypothesis based on models of possible objects (Fig. 7). These models are grouped in classes that incorporate pedestrians, cars and other road users; for each of these classes they define geometric and dynamic constraints. These constraints are used to limit the number of association hypothesis in order to make the algorithm feasible in real time. For each hypothesis, a quality and consistency check is performed; objects that pass these tests are supplied to another application. The main disadvantage of this approach has been described previously; the authors also propose that the number of hypothesis can be further limited if contextual knowledge is available; for instance, if the course of the road is known, objects beside the road can be ignored and a preferred direction for hypotheses can be used these could reduced the number of hypotheses to handle.

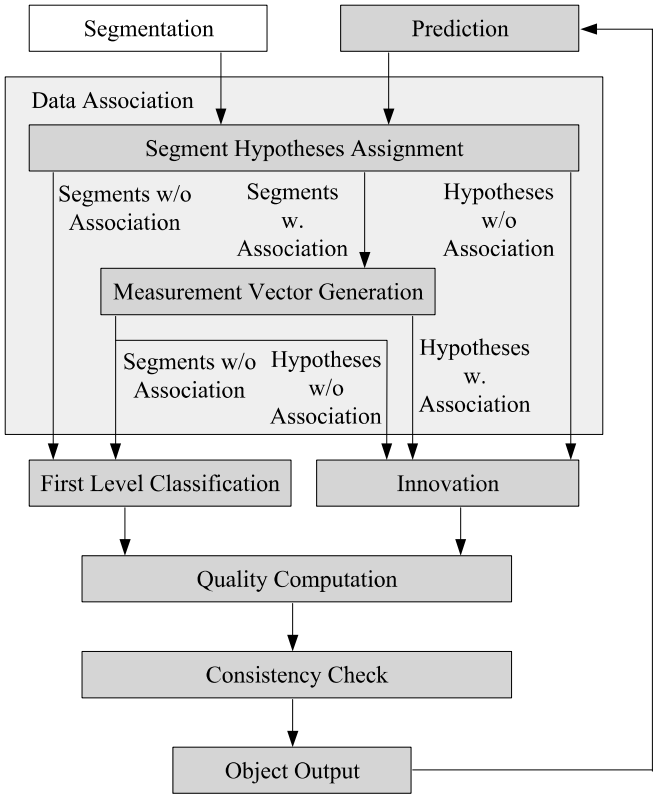

Fig. 7. Flow chart of the multi hypothesis approach used in (Streller and Dietmayer, 2004b).

In (Wang et al., 2007) the authors propose an algorithm that fuses the problem of simultaneous localization and mapping (SLAM) with detection and tracking of moving objects (DTMO). They propose that merging these two problems is beneficial since the SLAM problem can beneficiate from a correct identification of moving objects and the DTMO will also beneficiate with a more accurate pose estimation given by SLAM. In

their work they solve the tracking problem using a similar technique to SLAM without using any motion model; this alternative had some problems, as the authors noted, and they propose a modified algorithm that incorporates motion models to reduce computational time. In many applications, SLAM is of little use and is computationally demanding.

In Portugal there are also several studies in target tracking. In (Castro et al.) an algorithm for target tracking and feature extraction in indoors is proposed. Each laser scan is segmented using simple clustering and then features are detected using an adaptation of the Hough Transformation. To perform the tracking the center position of each object is estimated using the Kalman filter with a constant velocity model. A system intended to be applied in a collision avoidance system for outdoors vehicles is presented in (Mendes et al., 2004). Clustering is made using simple clustering but they implement a system that explicitly handles clustering errors for people legs, the system joints segmented groups that belong to the same person, and big obstacles like walls. Data association is performed using distance criteria as well as other features like dimension and orientation of objects. They perform classification of objects based on a voting scheme and multiple hypotheses. Tracking is preformed using Kalman filter and a constant velocity model just like in the previous work, the time to collision is calculated using the Kalman filter estimation. In (Monteiro et al., 2006) a system that combines laser and vision to perform tracking and classification is presented. In this system the laser is used for tracking and classification and the monocular camera is used only for classification. Laser scans are segmented using a Kalman based approach and the centroid of the group of laser points is used as the tracking feature. They perform data association in two steps, the first associates segments that may belong to the same objects in the current laser scan and the second associates current segments with tracked objects. The first association is performed using a combination of rectangular and ellipsoidal gates. The second association uses the result of the classification algorithm in regard to the object size and dynamic behavior.

Chapter 3

Tracking algorithm

3.1 Overview

Our algorithm is based on two main tasks, object construction and data association. The first task is responsible for the creation of high level objects from the laser scans. Basically, several techniques are applied to the raw laser scan in order to extract objects information; this is a bottom-up approach. These techniques compromise: preprocessing of the laser scan, clustering of points and data reduction. The preprocessing is intended to remove the laser sensor noise. Clustering creates groups of points that may belong to the same object. The data reduction algorithm reduces the amount of data to handle by fitting each object to a group of lines instead of a cluster of points; this is useful because the laser data will later be used by other algorithms. When working with 2D laser data we must assume that objects can only move in one plane, the laser scan plane. This assumption is generally true, but slopes or stairs can invalidate it; any motion in the perpendicular coordinate will create noise. Clustering is based on the assumption that objects can be separated based in distances between consecutive laser points. This assumption is not always valid because in the real world objects can lean to each other, when this happens, only a single cluster is created. To overcome this problem a fragmentation technique is proposed, it is described in 3.3.

In order to track objects, these must be maintained in time, so a list of objects to track is created. The data association phase is intended to associate objects that are currently visible to that ones that belong to our track list. This task is aided by motion prediction that is conducted using motion models and linear adaptive Kalman filter. The data association phase is critical for the correct tracking of objects, even more that the exact estimation of the object position and velocity. Wrong data association can lead to very bad velocity estimation, since it causes us to use measurements from different objects. We based our algorithm around this task and how it can be improved; we use a motion

based gating aided by heuristic rules that improve greatly the performance of the basic algorithm.

The source code for this project was developed in C++ programming language using Ubuntu 9.04 operating system. As mentioned before the algorithm is inserted into the ATLAS project, as such it was developed under the CARMEN standard (Montemerlo et al., 2003).

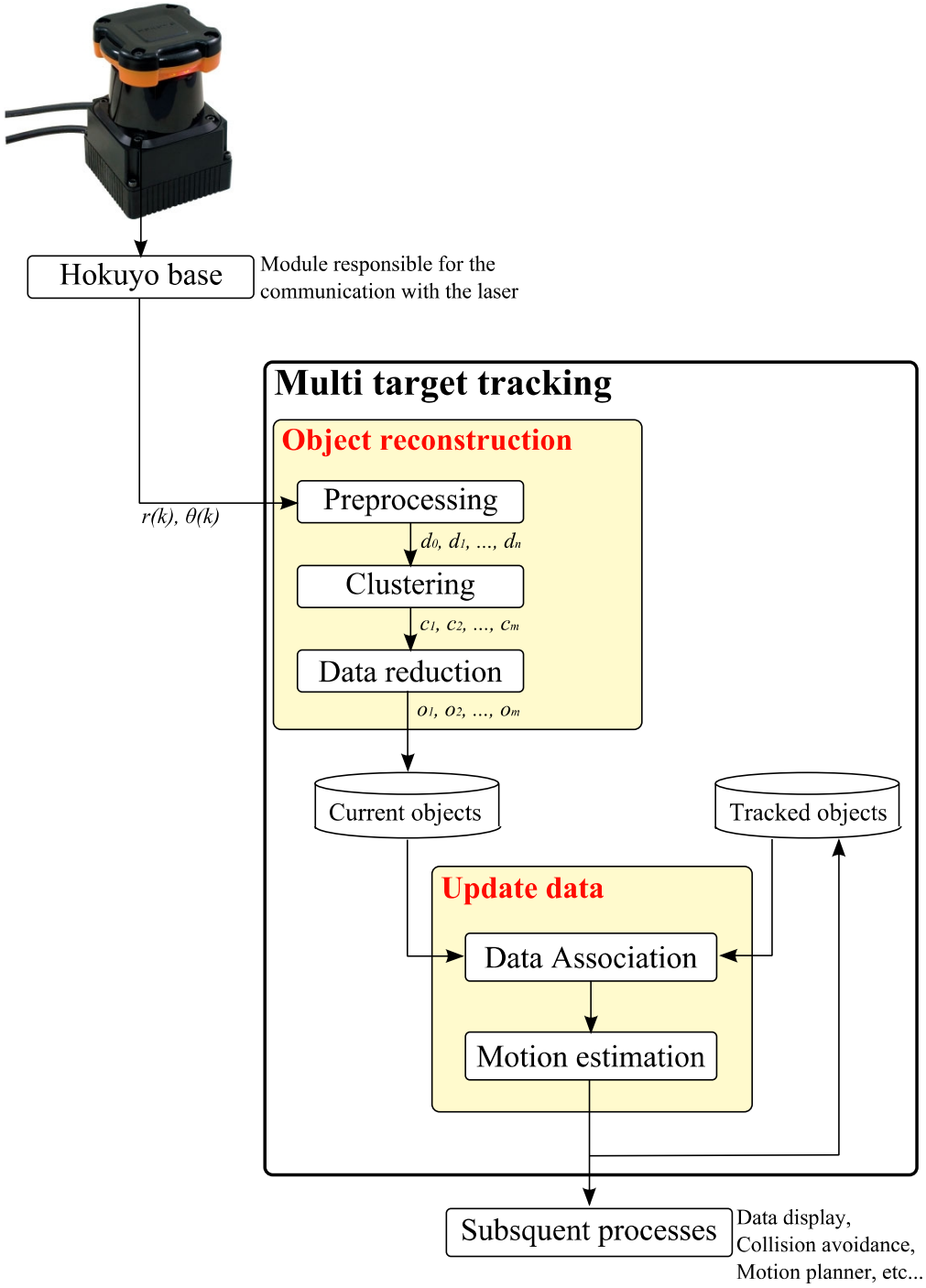

Fig. 8. Overview of the tracking algorithm.

3.2 Data preprocessing

Data is received from the sensor as a set of points in polar coordinates. This set is sorted by angle starting in the angle $\theta_0 = -45°$ and ending at $\theta_n = 225°$. The matrix d contains the group of n measurements received at a certain time (1).

$$d = \begin{bmatrix} r_0 & r_1 & \dots & r_n \\ \theta_0 & \theta_1 & \dots & \theta_n \end{bmatrix} \tag{1}$$

As stated previously, data obtained with the LRF is prone to high levels of noise; this noise can lead to inconstant clustering. Picture two objects at a fixed distance *D* of each other, if the cluster maximum distance *E* is less that *D* the two objects are correctly identified. Now imagine that the distance *D,* as measured by the laser range finder is $\tilde{D} = D \pm \varepsilon$, then it is possible that, for some value of ε, $\tilde{D}$ is less that *E*, when this happens the two objects are falsely identified as just one. The variable ε is time varying as so inconstant clustering may occur. To limit this effect a noise filter was applied. The filter was based in a moving average with heuristic additional rules that limited its influence in the responsiveness.

Heuristic Moving Average Filter

To perform the moving average filter a set of the N last data matrixes is maintained, this set is then averaged together (2) to create the current averaged data matrix. This average is recalculated every time new measurements are available.

$$\tilde{d}_k = \frac{1}{N} \sum_{i=k-N+1}^{k} d_i \tag{2}$$

This averaging removes most of the noise from the measurements but also compromises the responsiveness of the algorithm. To avoid this problem a simple heuristic rule is applied. For all data points in the data matrix (a point $d(i)$ is a column of the data matrix) if the newest data point is closer to the sensor more than a predefined threshold D we use the newest measurement instead of the average (3). This allows the filter to respond with great speed and makes the choice of N less relevant.

$$d(i) = \begin{cases} \overline{d}_k(i), & r_k(i) - \overline{r}_k(i) < D \\ d_k(i), & r_k(i) - \overline{r}_k(i) \geq D \end{cases} \tag{3}$$

The resulting filtered data matrix is denoted as d. Cartesian coordinates are then calculated for all data points using equations (4) and (5), and annexed to the polar coordinates.

$$x = r \times \cos(\theta) \tag{4}$$

$$y = r \times \sin(\theta) \tag{5}$$

r stands for the range value, θ for the angle, x and y are the Cartesian coordinates.

3.3 Clustering Process

The clustering step aggregates groups of points that are thought to belong to the same object based only on the distance between two consecutive points. This simple assumption brings some important problems. One big problem is the one already presented in previous chapter, inconstant clustering. Another problem is occlusion. Occlusion as a negative influence in clustering in many ways; one of the biggest problem is that every time a moving object (object A) starts to occlude another moving or stationary object (object B) this suffers from artificial movement (Fig. 9), the same happens when one moving object enters the occlusion zone of another object. This is due to the fact that a part of object B is being occluded by object A and so the centroid of the object B (as visible by the laser) moves in the direction of the remaining visible object. As the occlusion grows bigger the movement is continued and once the object is fully occluded the motion estimator and predictor will propagate the artificial movement, when the object B becomes visible again its centroid is by now very far from the predicted position of the object leading to miss detections. Another problem with clustering is that large objects can be segmented into smaller parts by smaller objects occluding them as sown in Fig. 10.

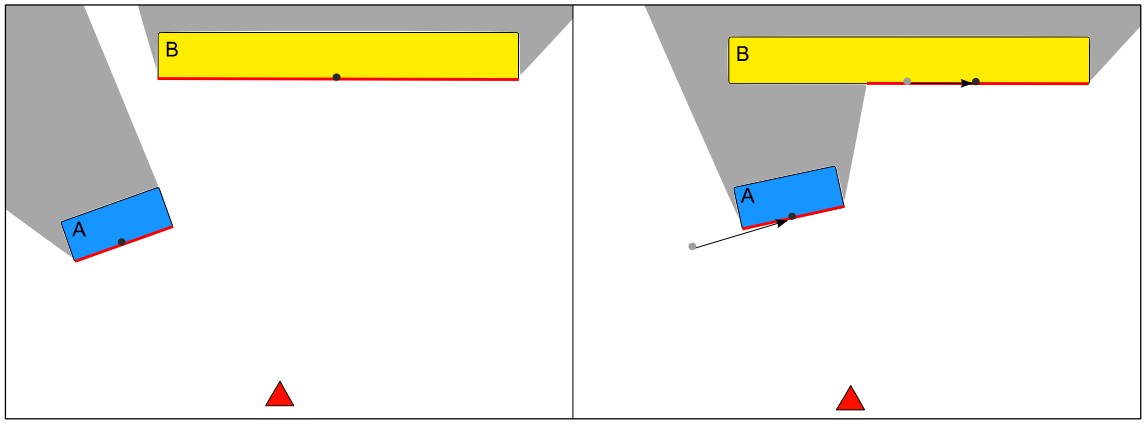

Fig. 9. Artificial movement caused by partial occlusions. Object B seems to move has it is occluded by A. The LRF is represented as a red triangle, occlusion zones are draw in grey, the visible part of the object is draw using a red line and the centre of each object is marked by a black dot.

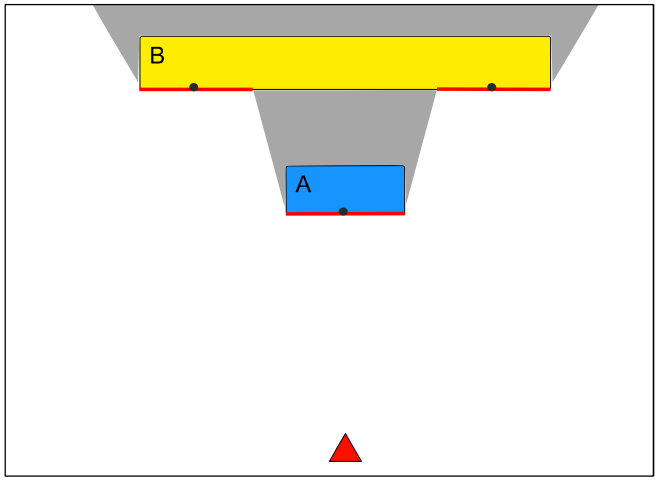

Fig. 10. Poor segmentation of objects, object B is segmented as two distinct objects instead of just one.

To overcome these problems we propose that occluded points should also be used to perform clustering along with new laser data (occlusion avoidance).

3.3.1 Occlusion avoidance

Our approach is based on detecting which points have become occluded and then create a new data matrix containing the latest data from the sensor and also the points which are known to be occluded from past iterations. Points p are classified as being occluded if in the new point in the same direction θ is closer to the sensor by more than a predefined threshold *OD*. The occluded data matrix is named *do* (6).

$$do = \{ p \in d_{k-1} : r_{k-1} - r_k > OD \} \tag{6}$$

Occluded points are kept until there's new information in the direction of that particular point. We assume that occluded points do not change while being occluded; this is not necessarily true because objects are free to move at will.

In order to aggregate the occluded points with the new data points we must know to which object the occluded points belong to; this is a requirement given that we will later perform typical gradient clustering on the joint data. The aggregation is performed using a first stage clustering that searches for break points in the filtered data matrix d, while coping the points to a new data matrix df, if a breakpoint is found, either at the start or end of a cluster, the algorithm tests if there's also a break point using occluded data do, if none is found that means that the occluded point belongs to this object (Fig. 11). When occluded points belonging to the current cluster are detected we cycle through the occluded data matrix in search of additional points, those points are added to the final data matrix, when a break point is detected we stop adding points and continue the

search in the filtered data matrix. Different search directions are used if the break point belongs to the start of the cluster or the end. In the end of this operation we obtain a final data matrix df that contains both the visible filtered data and the occluded data.

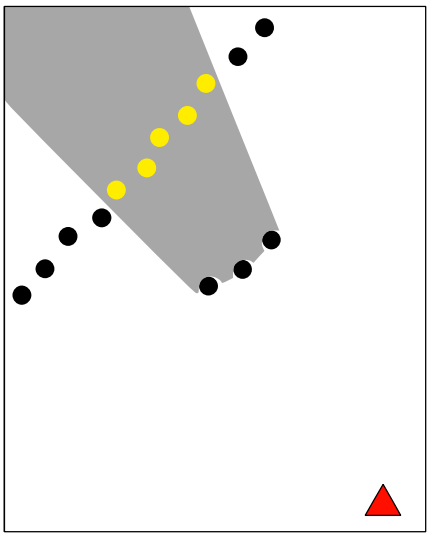

Fig. 11. Occlude point detection. Visible points of an object are drawn in black while occluded points are drawn in yellow. The final data matrix includes both. It can be seen that there's a breakpoint in the visible data in the crossing from the background object to the foreground object, this breakpoint does not exist if we use the occluded data.

3.3.1.1 Break point detection

Break points are detected based on the distance between two consecutive points using equation (7) (Dietmayer et al., 2001), $E_{i,i-1}$ denotes the Euclidian distance between two consecutive points and α the angular resolution of the laser. The constant C_0 allows and adjustment of the algorithm to noise and strong overlapping of pulses in close range.

$$E_{i,i-1} > C_0 + \sqrt{2 \times (\cos\alpha)} min\{r_i, r_{i-1}\} \tag{7}$$

3.3.2 Final clustering

Now that we have merged the visible points with the occluded points, we proceed to the final clustering, so we perform the same break point detection presented before but using the final data matrix df . Each time a break point is detected, the current cluster is terminated and a new one is created. One problem with clustering techniques is that if two objects come very close together they appear to merge (the merging distance depends on the breakpoint detection, equation (7)), for instance, if a person walks near a wall there may be times when no distinction can be made between the wall and the person made via clustering. To overcome this problem we decided to implement a

fragmentation technique; this technique consists of breaking large clusters in smaller ones based only on the cluster size; each time a cluster reaches a certain size, this cluster ends and a new one begins. The maximum cluster size was parameterized so that a person appears as just one cluster but anything bigger will be fragmented in two or more. Using this technique each cluster does not necessarily correspond to a different object; we propose that a higher level algorithm would join the clusters that belong to the same object (this algorithm was not developed). This algorithm could make use of clusters velocity and proximity between groups of clusters or even use geometric models. The direct advantage of the fragmentation is that when a person approaches a wall both of them will still be different clusters instead of just one (Fig. 12). Another advantage of this algorithm is that motion induced by bad clustering is much smaller and easier to manage; this is due to the smaller cluster size.

A cluster only contains information about its start position and end position in the data matrix. The end result of the clustering is a set of m clusters (8).

$$c = \begin{bmatrix} c_0 & c_1 & \cdots & c_m \end{bmatrix} \tag{8}$$

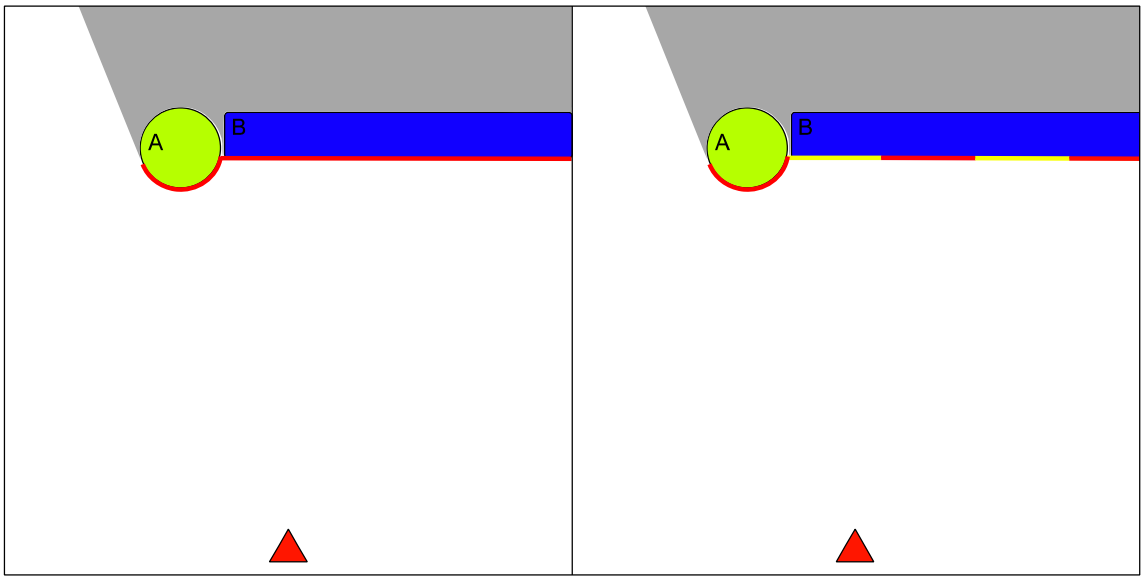

Fig. 12. Example of the cluster fragmentation. In the left image, object A is fused with object B due to their close proximity. In the right image large objects are fragmented, in this case object A is no longer fused object B, as it can be seen by the alternating colours of laser objects.

3.4 Data reduction

Given the fact that the laser data will be later used to perform tasks other than tracking, a data reduction technique was implemented. Objects in the real world are modeled as

groups of line segments; this approach is sufficient to all intended purposes in the subsequent tasks.

IEPF with mean variance

The process of transforming a cluster of points into a set of lines is based on iterative end-point fit (IEPF) presented by (Borges and Aldon, 2004). In this technique a set of points $P = \{p_0, p_1, \ldots, p_n\}$ is recursively split in two if a validation criterion is not met. In the first step a line is defined (in polar coordinates ρ, α) starting in the first point of the set p_0 and ending in the last p_n, then for each point the shortest distance to the line is calculated using expression (9). If the average distance $\tilde{d}$ is greater than a predefined threshold σ_{max} the set is split at the most divergent point and the process starts again to each one of the new sets. Polar coordinates are used to describe the lines.

$$d = \rho - x\cos\alpha - y\sin\alpha \tag{9}$$

The process is repeated for all m clusters, resulting in a set of m objects (10). These objects correspond to the currently visible objects; they contain information about the shape and position of the object but do not contain any information about past or future states.

$$O = \{o_0, o_1, \ldots, o_m\} \tag{10}$$

3.5 Motion estimation and prediction

In this work, objects' movement estimation and prediction was performed using an Adaptive Kalman filter. Estimation allows us to evaluate objects' velocities and true position while prediction allows us to perform localized search for possible matching reducing wrong associations. The Kalman filter formalization used was based on (Welch and Bishop, 1995).

3.5.1 Kalman filter

The Kalman filter addresses the general problem of trying to estimate the state $x_k \in \Re^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{11}$$

with a measurement $z_k \in \Re^m$ that is

$$z_k = Hx_k + v_k \tag{12}$$

The x_{k-1} vector represents the previous state; the u_{k-1} vector stands for the control vector. The random variables w_k and v_k represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, with normal probability distributions and with covariance Q_k and R_k respectively, note that both covariances can be time varying.

$$p(w) \sim N(0, Q_k) \tag{13}$$

$$p(v) \sim N(0, R_k) \tag{14}$$

The matrix *A* is the state transition matrix while *B* is the control matrix.

The Kalman filter equations can be grouped in two parts, the time update equations (predict) and the measurement update equations (correct). The first group is responsible for projecting forward the current state and the second group is responsible for incorporating the new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate. Each state prediction is followed by a subsequent correction.

To project the state estimation, the equation (15) is used; it makes use of the *a posteriori* estimation $\hat{x}_{k-1}$ from the previous time step, the state transition matrix *A* and the control matrix *B* and vector to calculate the *a priori* estimation for the current time step $\hat{x}_k^-$.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{15}$$

The error covariance is projected using the previous error covariance, the state transition matrix and the process noise covariance matrix (16).

$$P_k^- = AP_{k-1}A^T + Q_k \tag{16}$$

The measurement group of equations encompasses the *Kalman gain* calculation, the estimation update with the current measurement and the error covariance update step.

The *blending factor* or *Kalman gain*, *K*, of the filter is a $n \times m$ matrix that minimizes the *a posteriori* error covariance, it is calculated using equation (17).

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{17}$$

The *a posteriori* state estimation $\hat{x}_k$ is linear combination of the *a priori* estimate $\hat{x}_k^-$ and a weighted difference between an actual measurement z_k and a measurement prediction $H\hat{x}_k^-$ (18).

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{18}$$

The error covariance matrix is updated using the previous error, the *Kalman gain* and the measurement sensitivity matrix (19).

$$P_k = (I - K_k H)P_k^- \tag{19}$$

Adaptive Kalman

In order to overcome Kalman filter's heavy reliance on modeling accuracies, many adaptive methods have been developed.

Our approach is to adapt the white noise spectral amplitude based on the innovation error covariance. The innovation sequence is defined as

$$d_k = z_k - H_k\hat{x}_k^- \tag{20}$$

The real covariance of the innovation sequence can be approximated using

$$Cov\{d_k\} = E\{d_k d_k^T\} = \frac{1}{m}\sum_{j=0}^{m-1} d_{k-j} d_{k-j}^T \tag{21}$$

where m is the 'estimation window size'. This equation is only valid if the innovation sequence is assumed to be a time invariant process over the m steps.

The white noise spectral amplitude was calculated using the following equation

$$\sigma_{ak} = \sqrt{Cov\{d_k\}} \tag{22}$$

Residual noise is also calculated but only to evaluate the estimation performance, the residual sequence is defined as

$$\varepsilon_k = z_k - H_k\hat{x}_k \tag{23}$$

and the covariance of the residual sequence can be approximated using

$$Cov\{\varepsilon_k\} = E\{\varepsilon_k \varepsilon_k^T\} = \frac{1}{m}\sum_{j=0}^{m-1} \varepsilon_{k-j} \varepsilon_{k-j}^T \tag{24}$$

This adaptive implementation allows easier tracking of accelerating objects as well as a more stable estimation of the position of static ones, and makes the tracking performance less dependent of the error matrixes initialization. Other alternatives to adaptive Kalman estimation may be found in (Ding et al., 2006; Ding et al., 2007).

3.5.2 Motion models

In this work, two motion models were employed, a constant velocity model and a constant acceleration model. None of the models fully adjusts to the motion of a person; for instance, it's known that no object can have constant velocity from the beginning of

motion to the end, in the beginning the acceleration forces the velocity to increase and near the end the velocity decreases; attempts to use model human motion have been made in (Zhao and Shibasaki, 2005). The basic constant velocity model was superposed with white noise illustrating the highly time varying acceleration. The constant acceleration model was also superposed with white noise. Each object motion was modeled using two independent models, each model employed two Kalman filters, one per each coordinate (x, y) given that both measurements are uncorrelated this separation simplifies the process.

3.5.2.1 Constant velocity CV

Let s represent the position of an object and v its velocity, these variables correspond to a single coordinate either x or y; it is known that

$$\dot{s} = v \tag{25}$$

So that the position of an object at an arbitrary time t is given by

$$s_t = \int_0^t v dt \tag{26}$$

Assuming v as constant

$$s_t = v\Delta t + s_0 \tag{27}$$

In discrete time the equation above can be represented by

$$s_k = s_{k-1} + v_{k-1}\Delta t \tag{28}$$

$$v_k = v_{k-1} \tag{29}$$

In equation (29) it is clear that the model is of constant velocity, since the velocity of the current iteration is equal to that of the previous.

Using these two state equations, (28) and (29), one can derive the state vector (30) and transition matrix (31) as follows

$$x = \begin{bmatrix} s \\ v \end{bmatrix} \tag{30}$$

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{31}$$

Since only the position of objects is directly observable and it makes no sense to extract velocity from differences in positions since this task is performed by the Kalman filter, the measurement sensitivity matrix is

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{32}$$

Measurement error covariance matrix is defined by equation (33), σ_r^2 stands for the measurement error variance. The error variance is very hard to define accurately; the manufacturer of the laser sensor only provides the error variance for the range measurement and not angular error variance. Even if one knew these two variables the measurement error variance of the Kalman filter would not correspond to a conversion of these errors to Cartesian coordinates. The real error measurement depends on the tracking feature defined (which point of the object we are tracking); our feature (3.6.5) is highly dependent on the correct cluster definition and the error of each laser point that belongs to that particular cluster. The clustering error is very hard to estimate since it depends on a complex process and is influenced by factors like occlusions and false detections; on the other hand the error of each laser point depends on the polar coordinates errors but also on the preprocessing of these points (3.2). As so we opted to use an approximation of this error based on the laser manufacturer error and some empirical observation of the filter behavior with real data.

$$R = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix} \tag{33}$$

The process noise covariance matrix (34) was extracted from (Kohler, 1997) and it is time varying using the proposed adaptive Kalman method. Given that the motion model is a superposition of basic constant velocity model with white noise acceleration this equation presents this superposition.

$$Q_k = \frac{\sigma_{ak}^2 \times \Delta t}{6} \begin{bmatrix} 2\Delta t^2 & 3\Delta t \\ 3\Delta t & 6 \end{bmatrix} \tag{34}$$

3.5.2.2 Constant acceleration

In this model we assume constant acceleration (Wiener-process acceleration model) so the state equation results in the following tree equations; equation (35) for the position, (36) for the velocity and (37) for the acceleration, once again these equations only represent one coordinate, either x or y.

$$s_k = s_{k-1} + v_{k-1}\Delta t + a_{k-1}\frac{\Delta t^2}{2} \tag{35}$$

$$v_k = v_{k-1} + a_{k-1}\Delta t \tag{36}$$

$$a_k = a_{k-1} \tag{37}$$

In equation (37) it is clear that the acceleration is constant since the current acceleration is equal to the previous one. Using these equations (35), (36) and (37) the state vector results in (38) and the state transition matrix in (39).

$$x = \begin{bmatrix} s \\ v \\ a \end{bmatrix} \tag{38}$$

$$A = \begin{bmatrix} 1 & \Delta t & \Delta t^2 / 2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{39}$$

As in the constant velocity model, only the position is directly measured. So the sensitivity measurement matrix is as follows.

$$H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{40}$$

Once again the measurement error covariance is obtained from the sensor manufacturer and empirical observation. This error matrix suffers from the same issue as the constant velocity model, and once again the same solution was used.

$$R = \begin{bmatrix} \sigma_r^{\,2} & 0 & 0 \\ 0 & \sigma_r^{\,2} & 0 \\ 0 & 0 & \sigma_r^{\,2} \end{bmatrix} \tag{41}$$

In our implementation we assume that the acceleration derivative is an independent (white noise) process, $\dot{a}_k = w_k$. The corresponding process noise is presented in the following equation (Jilkov, 2009).

$$Q_k = \sigma_{ak}^{\,2} \begin{bmatrix} \frac{\Delta t^5}{20} & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix} \tag{42}$$

3.6 Data association

The most important task of any multi object tracking algorithm is the data association. Data association is the task of assigning the correct measurements to the each object. Our proposal is to use a motion based gating algorithm to perform this task, the algorithm searches for possible matches around the object predicted position. The basic assumption made here is that each object will maintain its motion in the next iteration, which is not always true. An object can suddenly stop or decide to turn very rapidly, and those fast changes in direction are the main cause of error in the algorithm; due to them the object that is closest to the target predicted location is not always the correct one, thus leading to miss associations. Another big cause of error is when two objects move together and if by any reason one of them becomes occluded, due to the object proximity and measurement errors, wrong association may occur.

To overcome some of these problems, the basic algorithm described was overlaid with some heuristic rules derived directly from observation of the algorithm with real data.

3.6.1 Object definition

In order to perform tracking, a list of tracked objects is maintained (43).

$$TO = \{to_0, to_1, \ldots, to_l\} \tag{43}$$

Each object of the list contains a group of variables that define its state (Table 1). Those variables indicate the current state of the object as well as past states. They include: object position (directly measured, estimated and predicted by the Kalman filter), velocity (only estimated), search area (used for object association) and current classification; the object previous positions are also stored and used to define the object path.

The object also has multiple motion models used in its motion estimation, associated with those models are the error sequences (innovation and residue) and their covariances. Each object also contains two timers that indicate the object life time and occlusion time. The next table resumes the mentioned fields.

Table 1. Object structure containing all data fields. This information is propagated in time for each object that is being tracked.

Object	
Position and velocity	Measured, z_k (only position) Estimated, $\hat{x}_k$ Predicted, $\hat{x}_{k+1}^-$
Search area	Ellipse major axes, ea Ellipse minor axes, eb
Classification	Motion based Visibility
Motion models	Constant velocity Constant acceleration
Errors	Innovation, d_k, and covariance, $Cov\{d_k\}$ Residue, ε_k, and covariance, $Cov\{\varepsilon_k\}$
Timers	Life time Occluded time
Path	Past estimated positions in x and y axis

3.6.2 Object association

In our work, the task of data association is making a correspondence between the objects being tracked TO and those that are currently visible O, and contain the new measurements z_k.

In the first iteration, all current objects O are directly added to the list TO; then, on each iteration, each object belonging to TO is searched for in current objects O. If a match if found, the TO object is updated with the corresponding current object information, if no match is found, we use the predicted state of object in TO as the current position measurement thus propagating the object motion, and at the same time we increment a missed detection counter. All current objects O that were not matched to any list objects are then added to the list TO. Objects are removed from the list when their missed detection counter is over a threshold $M_{Threshold}$. This is the basic principle of the association algorithm; several heuristic rules were developed to improve performance and are described in the following section.

3.6.3 Heuristic rules

In order to improve association of objects, several heuristic rules were added to the basic algorithm described above.

According to their velocity, objects are classified in two classes: moving objects and stationary objects. Objects classification can change if the object velocity changes, meaning that a stationary object can become a moving object and a moving object can become a stationary one. This classification is preformed with some level of hysteresis for stability.

One current object (from the O list) cannot be associated with two list objects (from TO list).

The missed detection threshold ($M_{Threshold}$) varies dynamically and depends on the object life-time; an object with a small life-time disappears in few iterations, objects with a long life-time lasts many iterations before being removed from the tracked list. This threshold saturates at a predefined maximum value.

For each TO object there exist two distinct exclusion zones, the first zone (*ezA*) is used to avoid adding object's fragments to the TO list and it is centered at the current object position; the second exclusion (*ezB*) zone avoids associations from nearby objects and is centered at the object predicted position, these two zones are visible in Fig. 13.

Many times, an object can fragment itself; one example is a person walking: when the laser is pointed below the hip, the two legs will sometimes appear as one object and sometimes as two objects; if the laser is put above that level, the same thing will happen but with the hands. To avoid creating false tracks, objects that appear within the *ezA* of one object are not added to the TO list.

If two persons walk together there will be a time when the person closest to the laser will occlude the other, one typical problem is associating the occluded object with the currently visible person or any of its fragments; the goal of the *ezB* is to avoid this problem by not allowing any associations other that the object it belongs to, within its radius.

Both exclusion zones are circular in shape but have different radius; the *ezA* radius is 0.8 m and the *ezB* radius is 0.5 m. This distinction will allow an existing object to get near another but will prevent the creation of new objects while they remain in close proximity of an existing.

3.6.4 Gating of objects

All list objects have an associated search area that is used to perform the gating of measurements. This area is shaped as an ellipse centered at the object predicted position as this is the object most probable location (Fig. 13).

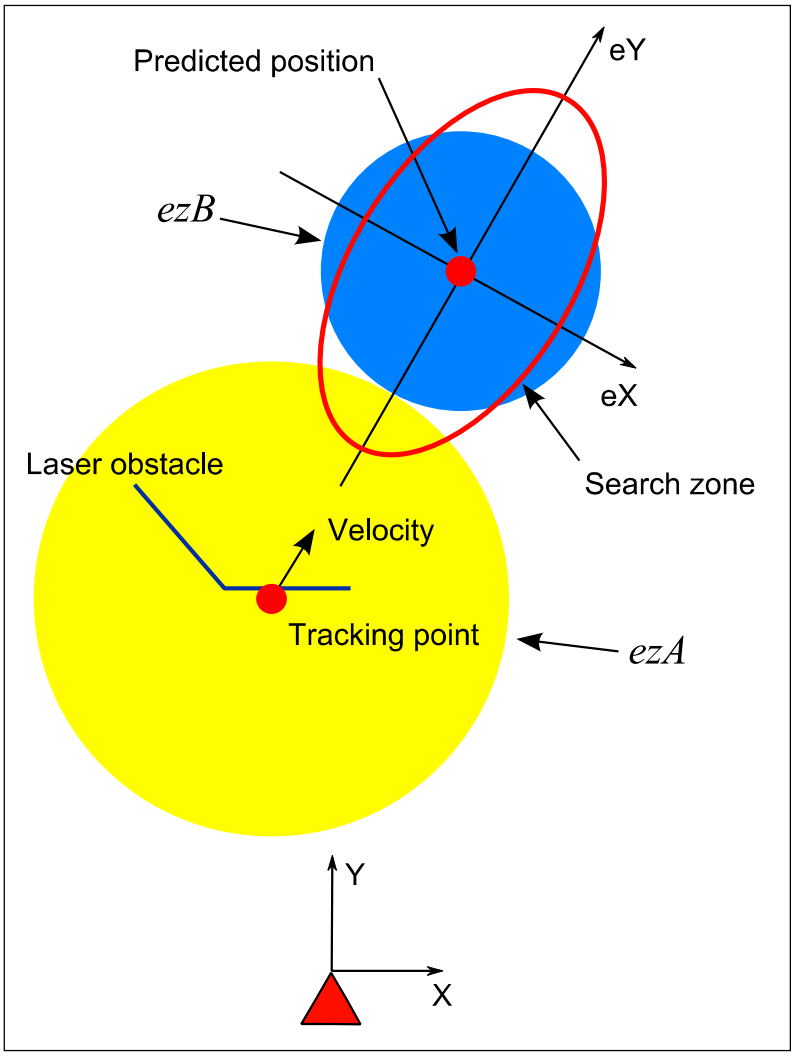

Fig. 13. Search area shape and important points. The exclusion zone A is drawn in yellow and centered at the current object position; the exclusion zone B is drawn in blue centered at the predicted position. The zones presented here correspond to two different time frames (for an easier interpretation), *ezA* belong to the k iteration while *ezB* to the k+1 iteration.

The ellipse is aligned with the object estimated velocity as extracted from the Kalman estimation. Ellipse axes lengths are dynamic calculated. Axes lengths are governed by equation (44) for the axis aligned with eY and (45) for the axis aligned with eX. Each axis has a default size ds that is equal to all objects; the axis length increases with the object size s by a size factor sf. The occluded time ot of the object also increases the axis length with an occluded factor of; both axis use the innovation error covariance

from the previous iteration d_{k-1} to dynamically change the length using an innovation factor if ; the axis aligned with eX depends also on the lateral error from the previous iteration by a lateral factor lf . The lateral error is the distance from the object position to the eY axis of the ellipse, it allows us to detect if the object is turning.

$$ea = ds + sf \times s + of \times ot^2 + if \times d_{k-1} \tag{44}$$

$$eb = ds + sf \times s + of + ot^2 + if \times d_{k-1} + lf \times le \tag{45}$$

The error factor is not equal in both axes and depends on the object velocity; if the object is stationary both axes have an equal if with value 4, meaning that the object can equally move in any direction and the search area must be a circle. When the object is moving the if of the axis aligned with eY is 4 while in the other axis the if is 0.5. While the axis aligned with eY is dominated by the covariance error the axis aligned with eX is dominated by the lateral error. By affecting the axes in different ways we can increase the size of the ellipse only where it is necessary, meaning that if an object is turning at a constant rate we don't increase the length of the ellipse but increase its width, if the object is accelerating but not turning we only increase the length of the ellipse and not its width.

The default size of the ellipse must be large enough to allow the object to start moving. Both models used cannot incorporate changes in the object behavior while occluded, so any deviations from the model will cause the object to be missed when it exits occlusion. To mitigate this problem, we increase the ellipse size according to the number of iterations whilst the object wasn't found; this gives room to some deviation from the model when the object is occluded. Maximum values have been parameterized for the length of the axes.

3.6.5 Tracking feature

For each object, only a single point is tracked, this point is intended to represent the object position with some invariance to the object rotation and shape deformation. In this work, the tracked point is defined as having, in polar coordinates, r equal to the minimum r of all object lines end and start points (total of n points) and θ equal to the mean of the lines points' θ. That is, for the n points in a given object, its r and θ are given by expressions (46) and (47). This point is somehow a more stable representation of the object position than the object centroid. The centroid position converges to the part of the object that as more points but laser points are not very well distributed along

the object, given that the part of the object that is closer to the laser will have an increased number of points as opposed the part far from the laser. This distribution does not represent the shape of the object. Our proposal of feature point also fails to represent correctly the object but it's somehow invariant to object rotation and shape changing. When an object rotates the extremities will change very slowly as well as the point that is closest to the laser; these two features do not present discontinuities.

$$r = min(r_1, r_2, \ldots, r_n) \tag{46}$$

$$\theta = mean(\theta_1, \theta_2, \ldots, \theta_n) \tag{47}$$

This way of defining the tracking point allows the point to be outside the object area, this is not relevant as the only requisite for this point is its continuity in time.

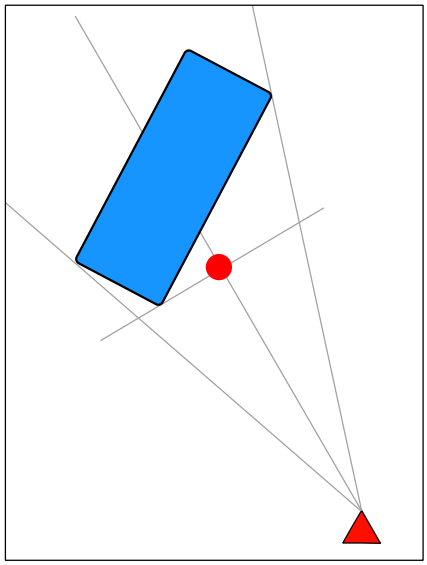

Fig. 14. The tracking point definition can lead to a point that is not inside the object area.

Chapter 4

Auxiliary tools

A very important part in any program development is debugging. In order to debug our algorithm real data was needed since no simulator was employed. Real data is hard to collect given that one must mount an experimental setup in a location with several moving targets; hence an algorithm to store data for later use was developed. This algorithm would record data sent by the laser module (Fig. 15) into a data file and later it would be possible to reproduce back the data to the modules requiring it. As previously mentioned this algorithm is inserted into the ATLAS project. In this project multiple programs (modules) exchange data (messages) using the CARMEM standard and IPC (Simmons, 1991). IPC allows a transparent and easy flow of messages between all modules.

The software consists of two modules, a Recorder module that does the message logging and a Player module that performs the playback. The software is able to log simple publish-subscribe messages and also more complicated query-server messages such as the ones exchanged through a shared memory (when efficiency is required).

Using this method, modules that subscribe the logged messages do not "know" that they are using logged data and not real time data, and thus allows for simulation with real data.

4.1 Recorder

This module is able to record messages sent by the sensors, not only laser but also cameras, inertial units, GPS, etc. Each log consists of several files, a header file and a variable number of data files. The header file contains information about which messages were received and were they were stored; it also stores generic information about the log, a more detailed list of the log fields is present bellow.

Table 2. Information contained in each log. There are two main types of data, generic data and messages data.

Log data	
Generic Information	**Messages**
Time and date	Timestamp
Computer name	Id
Start and end location of the log	Start position
Weather conditions	Message Length
List of sensors used	
List of messages types	

Some of the fields are automatically filled by the recorder while others must be input by the user. The log header file uses XML format to allow easy expansion and user interface. Each time a message is received the recorder marshals it into a byte array, following the IPC format of the message, and stores that byte array into a data file. Each message type has a different data file, this separation allows a user to select the desired message instead of using the entire log. Using the entire log can be difficult because of its extensive size, especially when images are recorded (the log can easily reach several gigabytes in size). The position in the data file where the message was recorded is saved in the header file to allow an easy retrieval of the stored data.

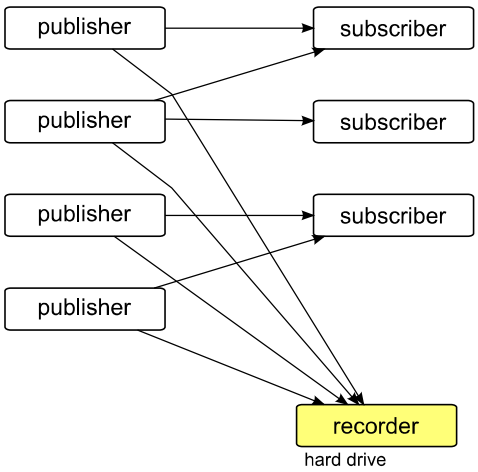

Fig. 15. Message flow to the recorder. Publisher modules send messages to the subscribers; the recorder also receives those messages and saves them into a file in the hard drive.

4.2 Player

To use the player module the user specifies which is the log to play; the player parses the XML header file and opens all the available data files. Each time a message is to be published the player reads the message from the corresponding data file using the start position and length of the message; then the byte array is unmarshalled into a message structure using the specific message type format and the message is finally published.

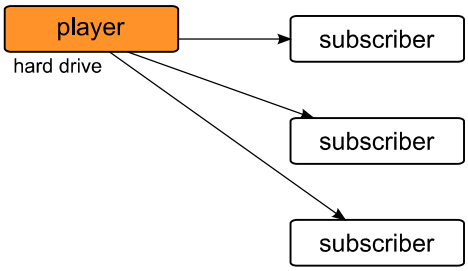

Fig. 16. Message flow from the player. The player reads messages from all the data files and publishes them to the subscribing modules.

To allow easy control of message publishing times, a graphic user interface was developed (GUI), Fig. 17. This GUI was developed using GTK+ in C++ programming language. It presents each message type stored in the log in a different time bar disposed vertically in parallel; each message is marked with a vertical line in the time bar against a white background. The player has control buttons that allow the user to start the playback, pause the playback, step a message forward and step a message backward. The step buttons jump to the next or previous message of the selected message type, all other messages for other message types that stand in the middle are instantly published disregarding their timestamp; the message type is selected by clicking on the corresponding time bar. The GUI also allows the user to set the speed at which the messages are published, a speed of one means that the messages are published at the same rate they were received, a speed lower than one means that messages are published at a slower rate and a speed higher than one means that messages are published at a faster rate, for instance a speed of two means that messages are played twice as fast as they were received. The GUI also allows the user to set the zoom level used to display the time bars, this is useful to analyze the logged messages. At the left of each time bar there's a check box, this check box allows the used to stop publishing that specific type of message. The current time is displayed using a vertical line with a small label on the bottom; this marker can be dragged inside the time bars to jump the log to another position. To jump the log a user can also click anywhere in the time bars to

move instantly the time. In order to identify which time bar corresponds to which message the user must leave the mouse cursor on top of the specific time bar, a text tooltip will pop up with the specific message type name. This tool also allows playing only a specific zone of the log (A-B, repeat feature); to specify the start (A) and end (B) time the user uses keyboard shortcuts.

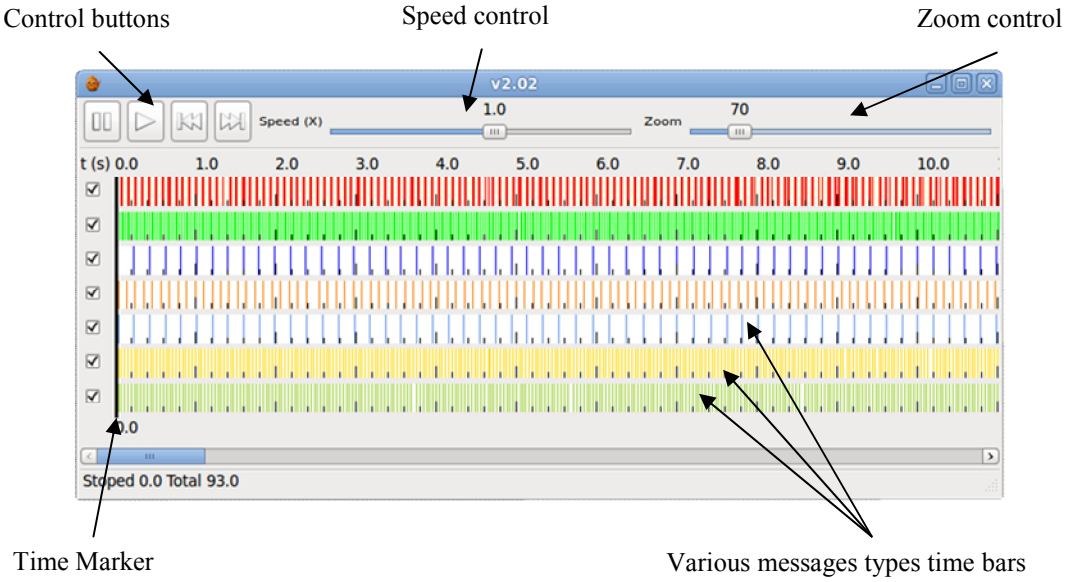

Fig. 17. GUI of the Player tool.

Chapter 5

Results

Several tests with different purposes were performed. The first test evaluates the Kalman filter performance on two distinct movements, linear oscillatory motion and circular motion. The second test evaluates the tracking of objects that move very near other objects, in this case a person walked leaned against a wall. The last experiment was intended to evaluate the performance of the algorithm is a real world test. In this case the laser was placed in a populated environment.

5.1 Kalman Filter estimation performance

In order to evaluate the performance of the Kalman estimation two experiments were conducted. In the first one the target moved away from the sensor in the Y axis and then back in an oscillatory motion. In the second experiment the target moved in circles around a fixed point. The first experiment was intended to test the algorithm against drastic speed changes, while the second one tested the algorithm against continued constant acceleration. In both experiments the models were tested using the same measurement error covariance and applying the proposed adaptive process noise covariance scaling method. Real data was used in both experiments; the target object was a person; so, in both experiments, the target does not follow exactly the stipulations above.

The Fig. 18 illustrates the oscillatory motion along the Y axis in the first experiment; position in the X axis will be omitted given that the motion along that axis is very small. In this graphic the measured position is superimposed with the estimated position using both motion models. It can be seen that the estimated position does not have the noise present in the direct measured positions. The position for the circular motion experiment is plotted in Fig. 19. As expected both axis present a similar sinusoidal motion. It can be seen that both models have a similar performance, the plotted curves are hard to distinguish because they superimpose most of the time.

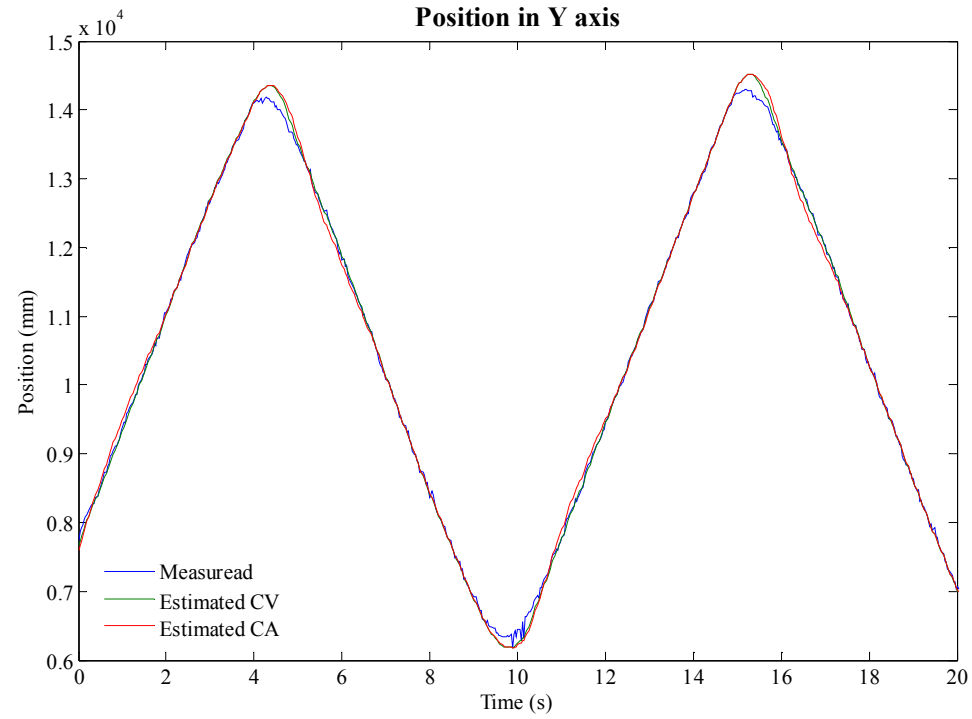

Fig. 18. Position of the target during the experiment in the Y axis. Measured and estimated positions are represented for the constant velocity and the constant acceleration model.

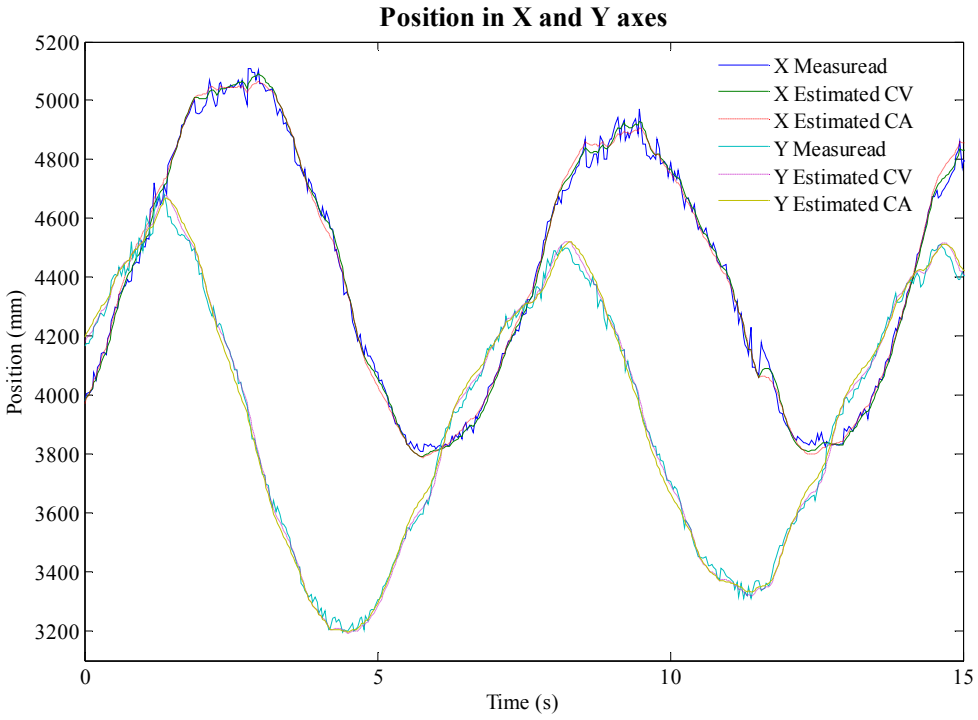

Fig. 19. Position in the X and Y axes during the second experiment.

5.1.1 Velocity estimation

Since the velocity was not measured directly only the estimated value is available. In the Fig. 20 the oscillatory motion can be clearly seen for the first experiment. The graphic belonging to the X axis is not present given that it doesn't contain any relevant information. It can be observed that the constant velocity model performs better that the constant acceleration when the object changes direction. It can also be seen that the target does not have constant velocity in between direction changes, the small oscillations in velocity correspond to the person small oscillations each time a step is made. In Fig. 21 we can observe the expected sinusoidal velocity in X and Y coordinates for the circular motion; it can also be noted that both models are able to catch the small oscillations in the walking motion.

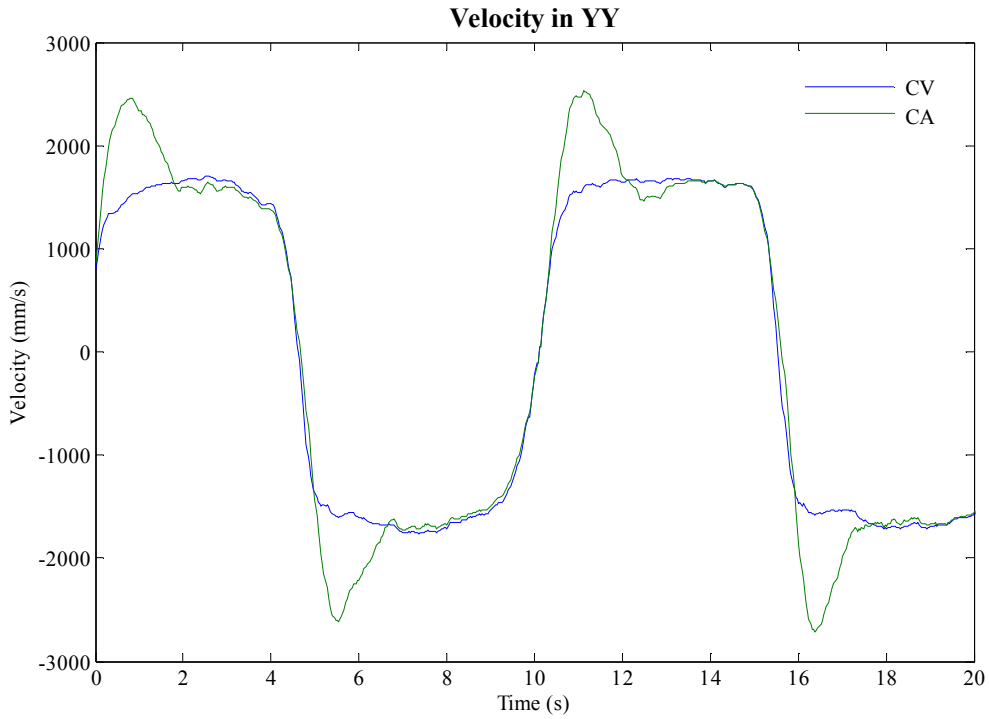

Fig. 20. Velocity along the Y axis in the first experiment; both models are able to follow the oscillatory movement but the constant acceleration model is slower to perceive changes in direction.

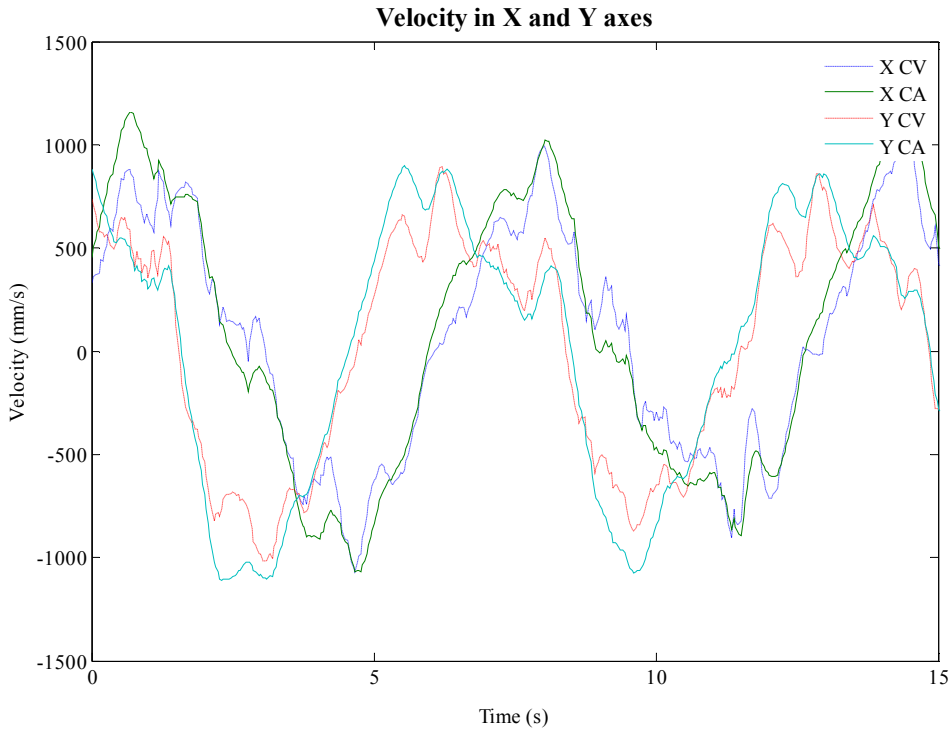

Fig. 21. Velocity in the second experiment along the X and Y axes.

5.1.2 Innovation sequence / Residue sequence

In the following graphics (Fig. 22) it can be observed, for the first experiment, that the constant velocity model performs slightly better that the constant acceleration model; the innovation mean of the CV model is -6.4 and standard deviation is 93.3, while the CA innovation model has a mean of -10.3 and standard deviation of 138.9. It can be clearly seen in the Fig. 22 that the CA model presents larger errors when the object changes direction. In the second experiment no significant differences were found between the two models.

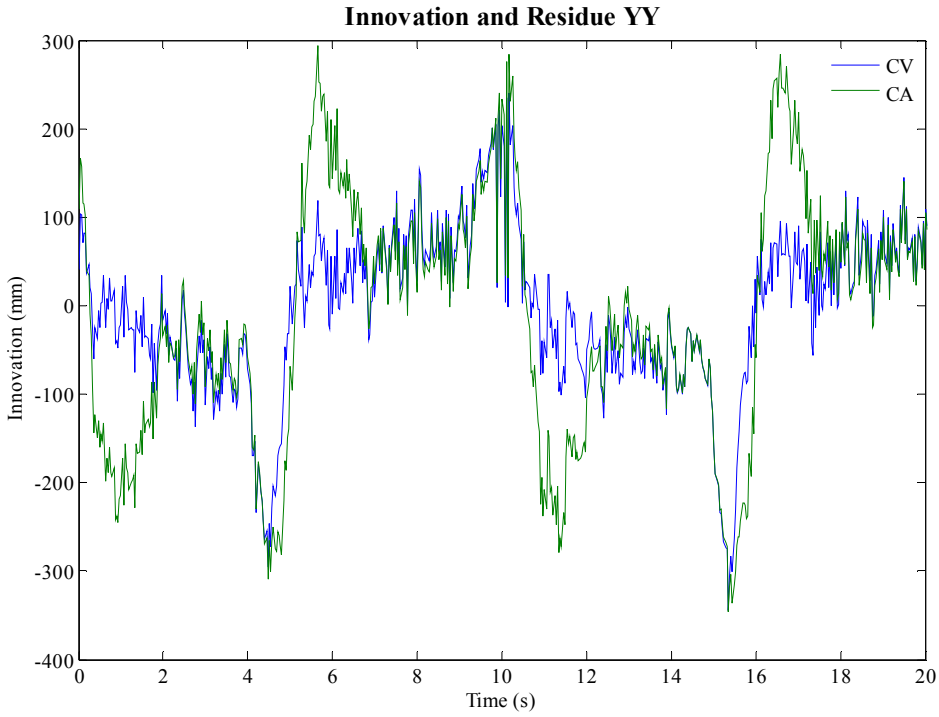

Fig. 22. Innovation error of both models during the first experiment.

5.1.3 Adaptive White Noise Amplitude estimation

Fig. 23 shows that the innovation covariance increases as the object changes its direction of motion; this fact is due to the acceleration required to change direction; these drastic changes are not modeled by any model and cause the predicted position of the object to diverge from the real position. This fact allows the usage of the innovation covariance to dynamically change the object search area size; the search area will increase when the object changes direction allowing for a bigger prediction error.

One curious fact that occurs in the CA model is a second spike each time the object changes direction; the first spike is the due to the inertia of the model that will attempt to continue in the same direction; the second spike occurs because of the feedback in the process noise covariance. While the error increases so does the process noise, in practice this means that the new measurements are more reliable than the previous, this will cause a big jump in the model estimated position; this jump will correct the error and even produce some overshoot that is visible as the second spike, this overshoot can also be seen in the velocity graphic (Fig. 20) but it is much more evident in this graphic. After a few iterations the value stabilizes to the correct value.

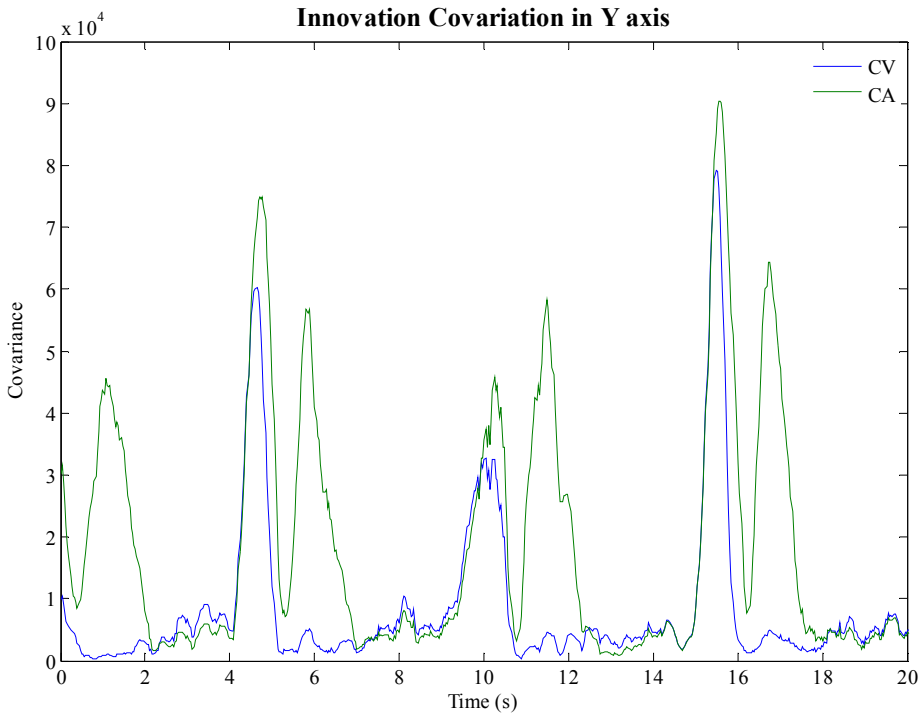

Fig. 23. Covariance in the Y coordinate using a window size of 10.

5.1.4 Kalman gains

In a time constant Kalman filter the gains are stationary but due to the adaptive nature of the algorithm implemented in our work, they are not constant.

The Kalman gains present here show that the filtering is stable.

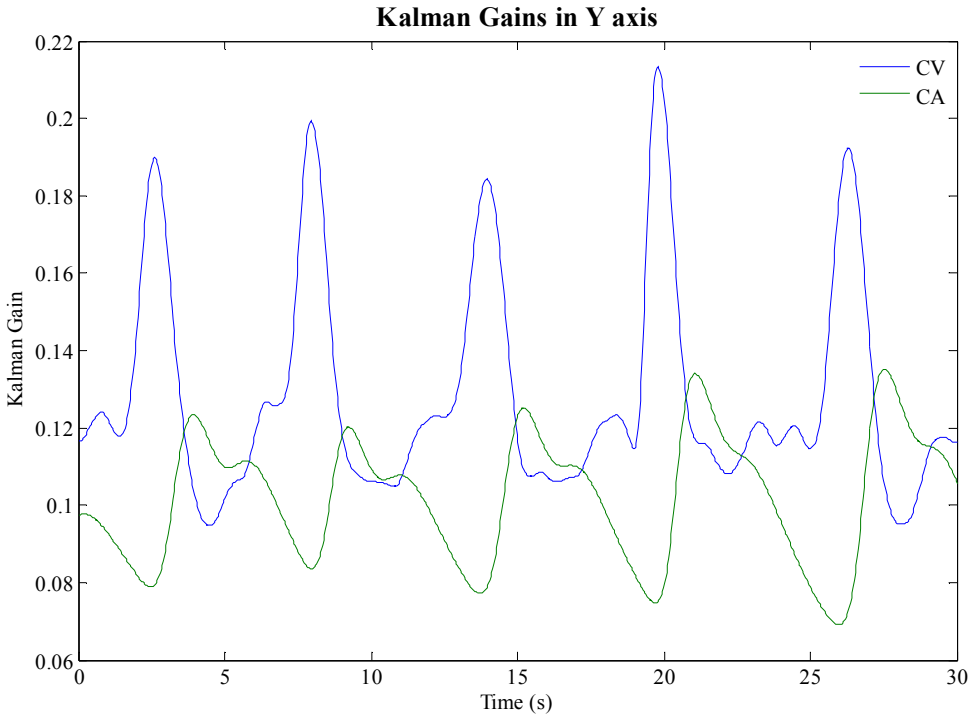

Fig. 24. Gains of the Kalman filters of both models in the Y axis for the first experiment. Only the first gain is presented here.

5.1.5 Conclusion

It was demonstrated that our constant velocity model performs better in fast direction changes that the constant acceleration model. In the circular experiment both models had a similar performance. Due to this conclusion and additional observations with real data it was decided to only use the constant velocity model in the following experiments.

5.2 Moving alongside a wall

In this experiment a person begun in a position far from the wall (Fig. 26) then started to move with the shoulder touching the wall (Fig. 25-A); after a while the person started to move with its back to the wall leaning on it (Fig. 25-B). Only the CV model was used.

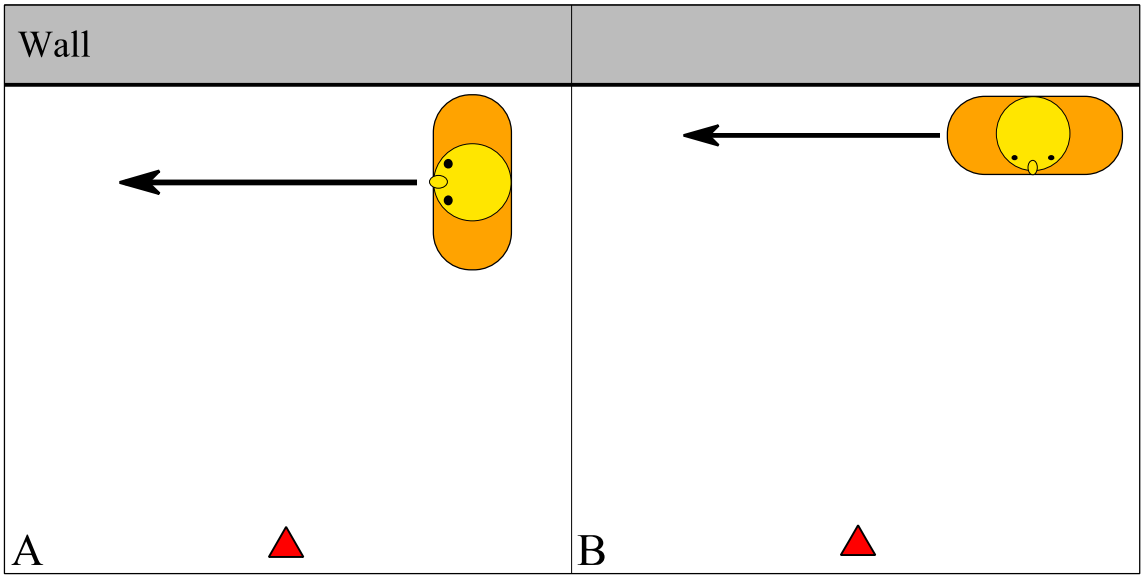

Fig. 25. This figure represents the two distinct movements that were made along a fixed object (wall).

This experiment is only possible due to the fragmentation technique employed, without it the person and the wall would be clustered together and one of the targets would be lost.

The algorithm presents some robustness to this kind of movement as it can be seen in Fig. 27. While the person walked with the should against the wall the target was always tracked correctly; when performing the second movement some wrong associations were made, but the algorithm maintained a high level of performance being the wrong associations only sporadic. The tracking of background objects (segments belonging to the wall) in this circumstances is very difficult and a high level of artificial movement as well as wrong associations were verified. In Fig. 27 it can be seen that more than half of the objects belonging to the wall were lost or out of place.

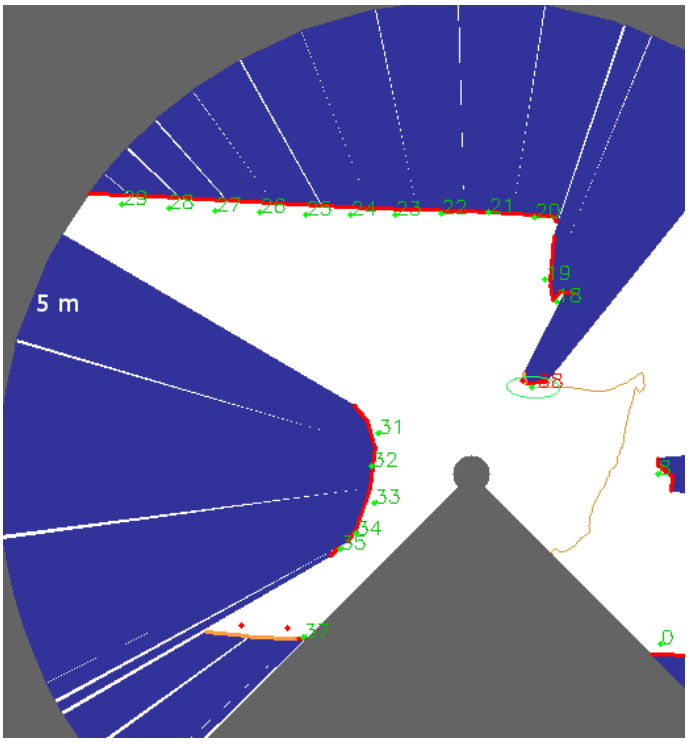

Fig. 26. Initial positions of objects in the scene. The target person has the id 38; ids from 18 to 29 correspond to the wall. Objects are marked as red lines, objects' occlusion areas are drawn in blue, ids are drawn in green and red (green for stationary objects and red for moving objects), object's tracks are in orange and out of range areas are drawn in grey. Moving objects search areas are drawn in green or purple if the object was not found. For static objects their search area was omitted.

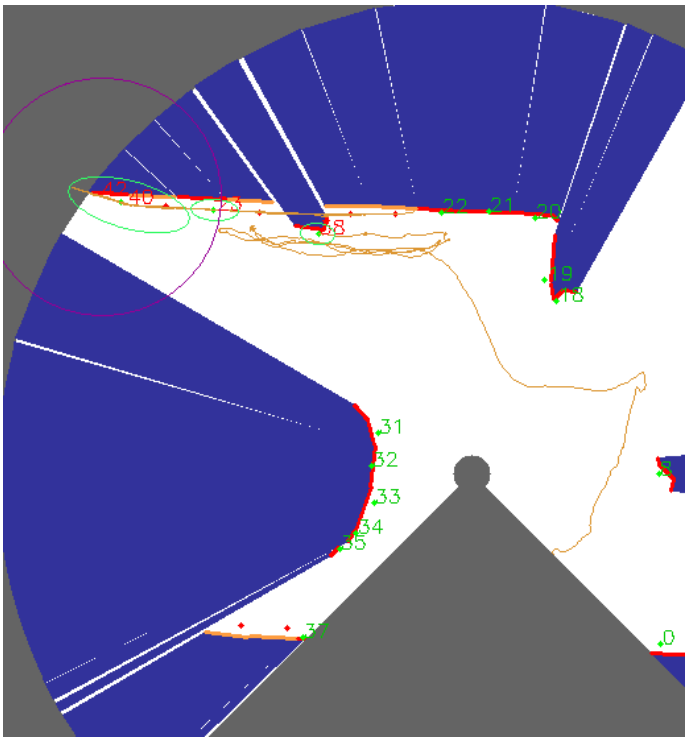

Fig. 27. In this frame the person started to move leaned against the wall. It can be seen that background objects (belonging to the wall) present some artificial movement and wrong associations but the foreground object (person) is still tracked correctly.

5.3 Algorithm performance

The goal of this experiment was to observe the behavior of the algorithm in a real world situation. To accomplish this goal, the laser sensor was placed in the front of a building entrance, observing two pathways (Fig. 28 and Fig. 29). Once again only the CV model was used.

Fig. 28. Photo of where the real world experiment took place.

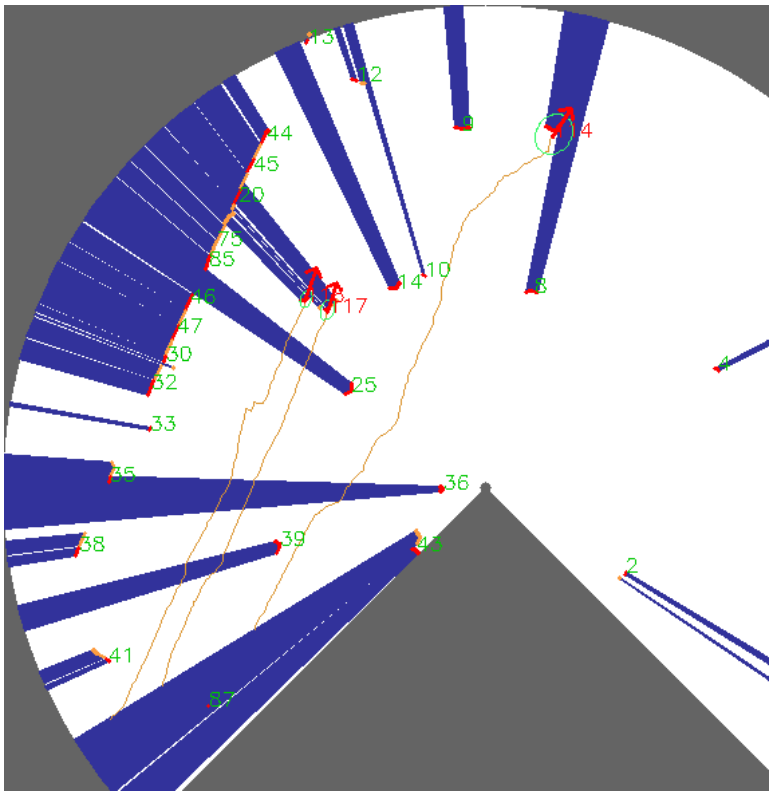

Fig. 29. Partial laser scan corresponding to the image in Fig. 28. Objects velocity is drawn with a red arrow.

In the scene, there were some pillars and trees that caused large occlusion in one of the pathways. During the experiment, people crossed the two pathways in both directions and some entered the building, one person was also observed picking up a bicycle and

moving away. The laser was configured for a maximum range of 15 meters and placed at a height of about 1 meter. The whole experiment was recorded with a camera in order to obtain ground truth information. The constant velocity motion model was used instead of the constant acceleration due to its higher performance.

Although there were only moving persons in our experiment, the targets were classified in two distinct types, A and B. Type A targets are single persons moving, type B are multiple persons moving together in the same direction and in close proximity. This distinction was done since the results for the two types of targets are fairly distinct.

In order to measure the performance of the tracking algorithm, the time each target was tracked was compared to the time that the target remained in sensor range, and also the number of tracking faults was measured for each target. Tracking faults compromise targets losses, target id switch, missed detections and false positives. An object is considered lost when it becomes occluded and it is not correctly identified once outside the occlusion area; in this case, an object may be tracked during all the time that it remained in scene (one id before being lost and a different id after), but one tracking fault is triggered due to the missed detection. Target ids switch in our experiment only occurred in type B targets; an id switch is detected when two targets switch their ids without switching place in real world.

During the experiment, 37 persons belonging to type A were tracked and 26 belonging to type B, making a total of 63 persons. The experiment lasted for approximately 17 minutes.

The overall tracking performance was good, as described next, and the algorithm performed significantly better for type A targets than for type B. Type A targets were tracked during 98.5% the time they remained in scene, and 5.4% (two targets) of the targets showed tracking faults; it was observed that the only fault that occurred for this type of target was target loss. These faults occurred due to large occlusion areas and objects that changed speed during occlusion (targets occluded while turning). The performance for type B targets was significantly worse, 89.9% of time tracked and 19.2% of the targets showed tracking faults. The main reason for the worse results is that when multiple persons move together the person that is closest to the laser sensor will occlude the others during a long time (when the targets move perpendicular to the laser, Fig. 30); during this large occlusion time it is very difficult to track the occluded persons since they only appear sporadically. If the two persons perform a curve, this occlusion will be even more dramatic because the persons will not possess a constant

velocity during the occlusion; this behavior typically leads to the loss of one of the targets. In the experiment, targets suffered different kinds of faults being the most common the target loss.

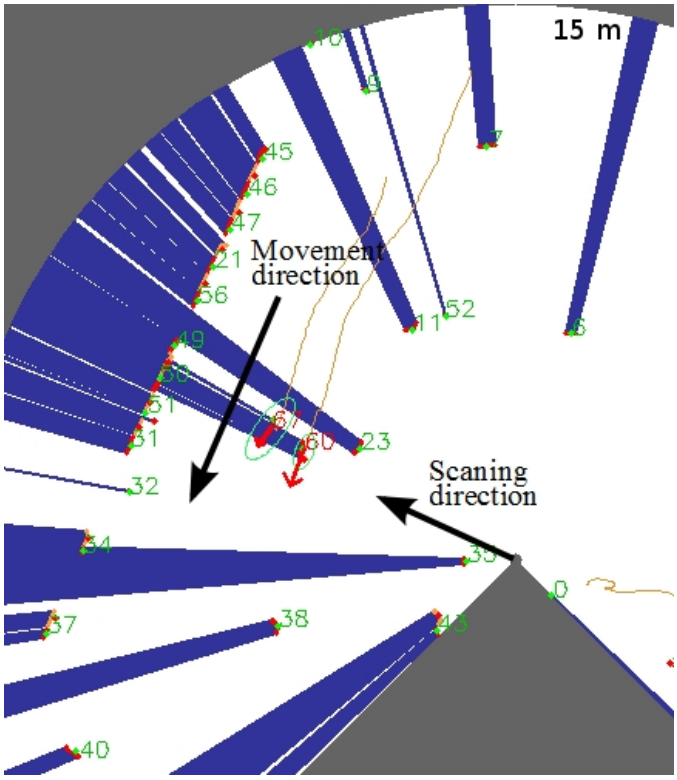

Fig. 30. Long occlusions caused by perpendicular motion of objects. In this case object 67 is occluded by object 60 and only appears sporadically.

Another phenomenon that occurred with multiple persons was the creation of false objects; this was due to clustering errors given the close proximity between the targets. Clustering errors were also seen in single person, and were due to the height at which the laser was placed, near the hip; at this height, peoples' hands, or other objects that they may carry, sometimes create new objects; this problem was overcome with the heuristic rule that prevents objects from being tracked if there is already a tracked object in the vicinity (exclusion zone *ezA*). The results of the experiment are summarized in Table 3 for both target types.

Table 3. Performance of the tracking algorithm during the experiment.

Type	**No. of targets**	**% time tracked**	**% objects with tracking faults**
A	37	98.5	5.4
B	26	89.9	19.2

It was observed that the person that picked up the bicycle, and moved away rapidly, was successfully tracked for the entire path while in range; this was only possible due to the

adaptive Kalman estimation. Under normal estimation, the white noise amplitude required to track the bicycle could destabilize less dynamic objects; by using the proposed adaptive technique, an object with small acceleration (low innovation covariance) have small white noise amplitude and very dynamic object (high innovation covariance) have large white noise amplitude.

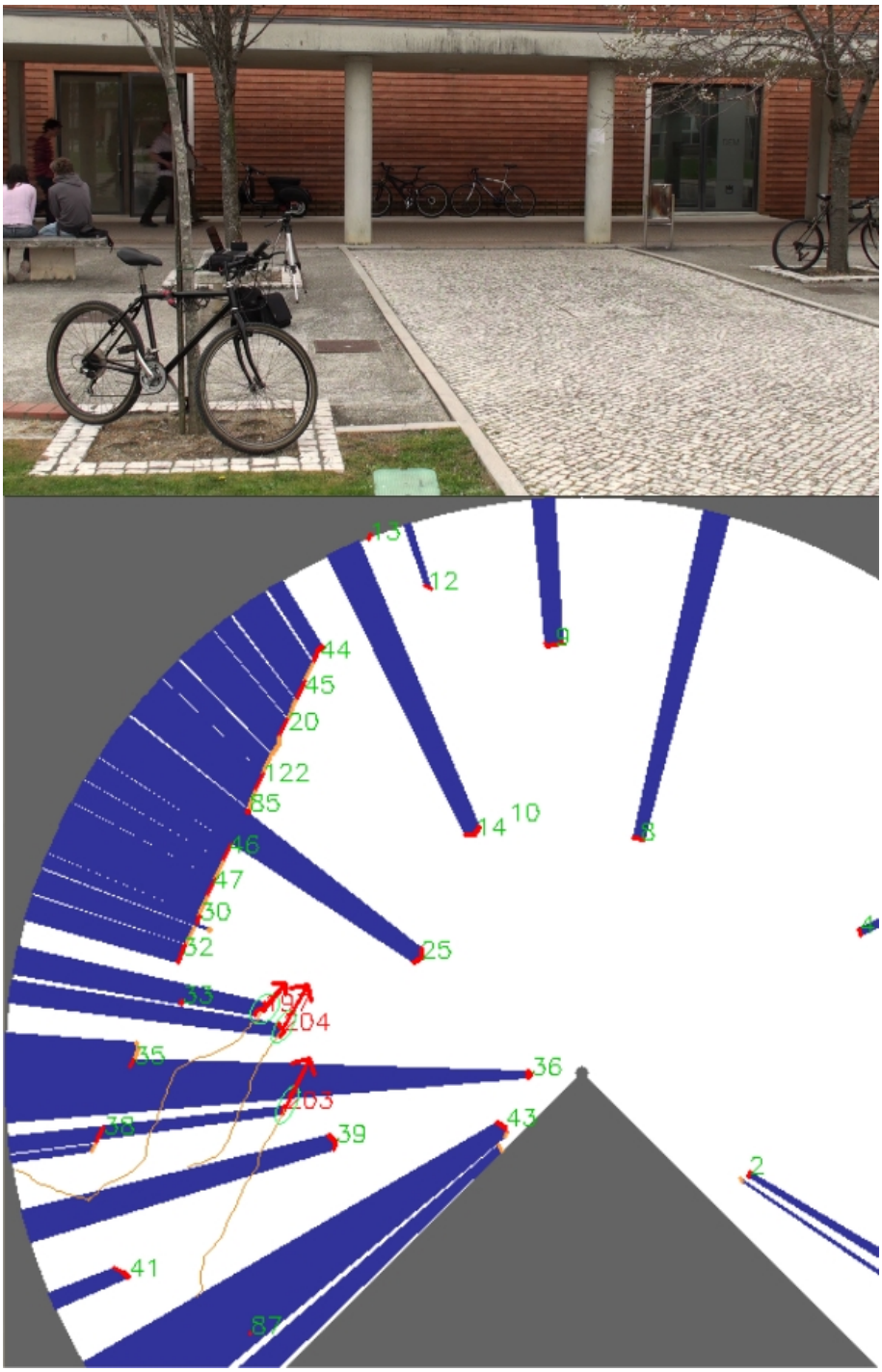

Fig. 31. Extract of the experiment. In this sequence tree persons move together being periodically occluded by the pillars and by one another.

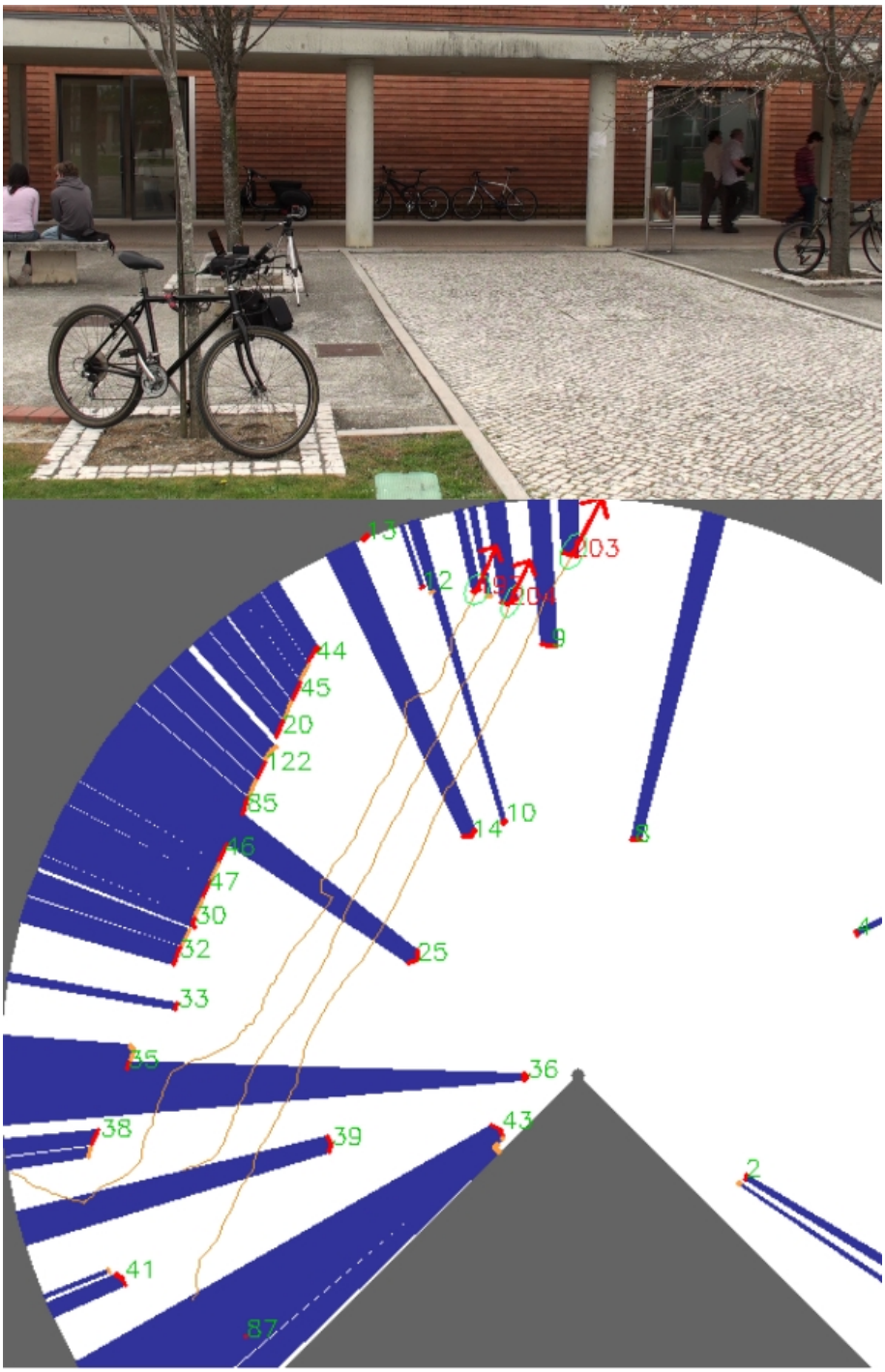

Fig. 32. The person closest to the laser overtook the over two and is now exiting the scene. During overtake the person 203 occluded the other two, the person 204 occluded the 197 and all tree where occluded by static objects.

Chapter 6

Conclusions and Future Work

A laser based object tracking algorithm was developed and successfully implemented. The algorithm uses multi stage clustering techniques supported by additional heuristic rules and adaptive Kalman estimation to correctly track objects even when occlusion happens. As long as objects' motion remains under the model conditions even long occlusion times may be well managed.

Although the basic operation of the algorithm is relatively simple and performs rather poorly by itself, it was demonstrated that a very good performance can be achieved by augmenting the algorithm with simple heuristic rules. An adaptive Kalman estimation technique was also employed and demonstrated to improve the algorithm robustness to deviations from the basic motion model and stabilization of objects.

To fully test the algorithm, a real word situation was used. The experiment was conducted in a populated unstructured environment and good performance was achieved.

Multiple persons moving together still pose some problems that could be overcome with the integration of additional lasers or other kind of sensors, such as cameras. Two auxiliary tools were also developed; these tools allowed the record and playback of data messages. They were widely used in this work to perform debug and test the algorithm, their usage proved to be invaluable as these tools are also used by other coworkers in the development group to log and record messages from other kind of sensors.

The algorithm was not tested with the sensor in movement (ego-motion). The ego-motion can bring several problems to the target tracking; using only the laser data it is hard to distinguish the real velocity of the targets from our ego-motion. This problem could be overcome if we know our own velocity (using, for instance, an inertial measuring device). We assume that if we have precise ego-motion compensation the tracking problem will be fairly equal to the static problem and our algorithm will stay

valid. This ego-motion compensation can even be assisted using the laser data, by means of frame to frame localization.

One of the main issues of target tracking is the data association; future work will be the implementation of more complex association techniques and compare their performance and computational cost with the used in this algorithm. Some of the most promising techniques are the MHT and the joint probabilistic data association filter (JPDAF).

Another important step in our algorithm is the clustering process. Our fragmentation technique helps making clustering less of a problem but introduces another problem: a single object is represented by multiple clusters. We propose that a higher level algorithm would join the clusters that belong to the same object. In order to provide reliable velocity estimation when tracking road vehicles, our tracking feature (single point) needs to take into account the geometry of the object being tracked. We propose that the algorithm that would join clusters together could use geometric objects models to do so.

Although motion estimation is not the primordial issue in target tracking, we believe that improvements on this could provide an easier data association. None of the two motion models presented is able to fully capture the extremely non linear motion of a person; as so we can evolve the motion model to be more complex (constant turn rate models, human motion models) or even incorporate interacting multiple models (IMM). The estimation algorithm can also be upgraded if needed to a non linear estimator like the extended Kalman filter or even to a non Gaussian estimator like particle filters.

Laser data is fairly poor; because of this the tracking is very difficult. If we use extra sensors, lasers or cameras, the tracking performance could be improved. The usage of additional laser units would reduce occlusions leading to an easier data association, and the use of cameras could, for instance, help data segmentation.

References

Andersen, Hans-Erik, Robert J. McGaughey and Stephen E. Reutebuch (2005). "Estimating forest canopy fuel parameters using LIDAR data." *Remote Sensing of Environment* **94**(4): 441-449.

Arras, K. O., S. Grzonka, M. Luber and W. Burgard (2008). Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. *IEEE International Conference on Robotics and Automation (ICRA), Pasadena, CA, USA*.

Blackman, S. S. and R. Popoli (1999). *Design and analysis of modern tracking systems*, Artech House Norwood, MA.

Borges, G. A. and M. J. Aldon (2004). "Line extraction in 2D range images for mobile robotics." *Journal of Intelligent and Robotic Systems* **40**(3): 267–297.

Castro, D., U. Nunes and A. Ruano Feature extraction for moving objects tracking system in indoor environments. *Proc. 5th IFAC/euron Symposium on Intelligent Autonomous Vehicles***:** 5–7.

Cui, Jinshi, Hongbin Zha, Huijing Zhao and Ryosuke Shibasaki (2007). "Laser-based detection and tracking of multiple people in crowds." *Computer Vision and Image Understanding* **106**(2-3): 300-312.

Cui, Jinshi, Hongbin Zha, Huijing Zhao and Ryosuke Shibasaki (2008). "Multi-modal tracking of people using laser scanners and video camera." *Image and vision Computing* **26**(2): 240-252.

Dias, P., M. Matos and V. Santos (2006). "3D reconstruction of real world scenes using a low-cost 3D range scanner." *Computer Aided Civil and Infrastructure Engineering* **21**(7): 486.

Dietmayer, K. C. J., J. Sparbert and D. Streller (2001). Model based object classification and object tracking in traffic scenes from range images. *Proceedings of IV 2001, IEEE intelligent vehicles symposium***:** 2–1.

Ding, W., D. J. Wang and C. Rizos (2006). Stochastic modelling strategies in GPS/INS data fusion process. *Symposium on GPS/GNSS*.

Ding, W., J. Wang and C. Rizos (2007). "Improving covariance based adaptive estimation for GPS/INS integration." *The Journal of Navigation* **60**(3): 517–529.

Fod, A., A. Howard and M. J. Mataric (2002). Laser-based people tracking. *Proc. of the IEEE International Conference on Robotics & Automation*.

Fuerstenberg, K. C., K. C. J. Dietmayer and V. Willhoeft (2002). Pedestrian recognition in urban traffic using a vehicle based multilayer laserscanner. *IEEE Intelligent Vehicle Symposium, 2002*. **1**.

Jilkov, V. P. (2009). "A Survey of Maneuvering Target Tracking: Dynamic Models." *Relation* **10**(1.56): 7880.

Kobilarov, M., G. Sukhatme, J. Hyams and P. Batavia (2006). People tracking and following with mobile robot using an omnidirectional camera and a laser. *Proceedings of the IEEE International Conference on Robotics and Automation***:** 557–562.

Kohler, M. (1997). "Using the kalman filter to track human interactive motion-modelling and initialization of the kalman filter for translational motion." *University of Dortmund, Germany*.

Mendes, A., L. C. Bento and U. Nunes (2004). Multi-target Detection and Tracking with a Laserscanner. *Proc. of 2004 IEEE Intelligent Vehicles Symposium*.

Monteiro, G., C. Premebida, P. Peixoto and U. Nunes (2006). Tracking and classification of dynamic obstacles using laser range finder and vision. *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Montemerlo, M., N. Roy and S. Thrun (2003). Perspectives on standardization in mobile robot programming: The Carnegie Mellon navigation (CARMEN) toolkit. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems***:** 2436–2441.

Oliveira, Luciano, Urbano Nunes, Paulo Peixoto, Marco Silva and Fernando Moita "Semantic fusion of laser and vision in pedestrian detection." *Pattern Recognition* **In Press, Corrected Proof**.

Oliveira, M., P. Stein, J. Almeida and V. Santos (2009). Modular Scalable Architecture for the Navigation of the ATLAS Autonomous Robots. *9th Conference on Autonomous Robot Systems and Competitions*. Castelo Branco.

Petrovskaya, Anna and Sebastian Thrun (2009). "Model based vehicle detection and tracking for autonomous urban driving." *Autonomous Robots* **26**(2): 123-139.

Premebida, Cristiano, Oswaldo Ludwig and Urbano Nunes (2009). "LIDAR and vision-based pedestrian detection system." *Journal of Field Robotics* **26**(9): 696-711.

Reid, D. B. (1978). An algorithm for tracking multiple targets. *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*. **17**.

Reutebuch, S. E., H. E. Andersen and R. J. McGaughey (2005). "Light detection and ranging (LIDAR): an emerging tool for multiple resource inventory." *Journal of Forestry* **103**(6): 286–292.

Samuels, M. A., S. W. Patterson, J. A. Eppstein and R. L. Fowler (1992). Low-cost hand-held lidar system for automotive speed detection and law enforcement. *Proceedings of SPIE*. **1633:** 147.

Simmons, R. (1991). *The inter-process communication (IPC) system*, Tech. report, 1991.

Stockdonf, H. F., A. H. Sallenger Jr, J. H. List and R. A. Holman (2002). "Estimation of shoreline position and change using airborne topographic lidar data." *Journal of Coastal Research*: 502–513.

Streller, D. and K. Dietmayer (2004a). Multiple hypothesis classification with laser range finders. *Proceedings of the IEEE Intelligent Transportation Systems Conference***:** 195–200.

Streller, D. and K. Dietmayer (2004b). Object tracking and classification using a multiple hypothesis approach. *IEEE intelligent vehicles symposium*.

Streller, D., K. Furstenberg and K. Dietmayer (2002). Vehicle and object models for robust tracking in traffic scenes using laser range images. *Intelligent Transportation Systems***:** 118–123.

Wang, C. C., C. Thorpe, S. Thrun, M. Hebert and H. Durrant-Whyte (2007). "Simultaneous localization, mapping and moving object tracking." *The International Journal of Robotics Research* **26**(9): 889.

Welch, G. and G. Bishop (1995). "An introduction to the Kalman filter." *University of North Carolina at Chapel Hill, Chapel Hill, NC*.

Yilmaz, A., O. Javed and M. Shah (2006). "Object tracking: A survey." *ACM Computing Surveys (CSUR)* **38**(4): 13.

Zhao, H. and R. Shibasaki (2005). "A novel system for tracking pedestrians using multiple single-row laser-range scanners." *IEEE Transactions on Systems, Man and Cybernetics, Part A* **35**(2): 283–291.