Multi-sensor data fusion for representing and tracking dynamic objects

Pawel Kmiotek

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I ELEKTRONIKI
Katedra Informatyki

UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD
Laboratoire Systèmes et Transports

Doctor of Philosophy Dissertation

MULTI-SENSOR DATA FUSION FOR REPRESENTING AND TRACKING DYNAMIC OBJECTS

MGR INŻ. PAWEŁ KMIOTEK

Supervisors:
Professor Yassine Ruichek
Professor AGH dr hab. inż. Krzysztof Cetnarowicz

Kraków, 2009

Contents

Acknowledgements

I would like to express my deep gratitude to my supervisors Professor Yassine Ruichek from the Systems and Transportation Laboratory at the University of Technology of Belfort-Montbéliard in Belfort, and Professor Krzysztof Cetnarowicz from the Faculty of Electrical Engineering, Automatics, Computer Science and Electronics at the AGH University of Science and Technology for their help and academic care during the realisation of this thesis. I am grateful for sharing their knowledge and experience with me and for their great commitment.

My thanks to reviewers Professor Jean-Charles Noyer from Université du Littoral Côte d'Opale and Professor Olivier Colot from Université Lille 1 for accepting the trouble of reviewing the cotutelle thesis dissertation, and for valuable and constructive comments.

I would also like to thank my colleagues from the Systems and Transportation Laboratory at the University of Technology of Belfort-Montbéliard in Belfort, Ms Cindy Cappelle, Mr Cyril Meurie and Mr Frédérick Zann for discussing the issues included in this thesis, for help in realising my ideas, and finally for the feedback which was of significant help in completing the dissertation.

Thanks to Ms Jolanta Lepiarczyk and Ms Ariane Glatigny for their great administrative support.

The special thanks go to my family for their moral support and sacrifices, which allowed me to materialize my dreams.

Chapter 1

Introduction

This dissertation is the final report of the works performed during my PhD studies. The studies were a part of the collaboration between the University of Technology of Belfort-Montbéliard (UTBM) and the University of Science and Technology in Krakow (UTS-AGH). It took the form of PhD "Cotutelle" scholarship.

The subject of the thesis lies within the scope of the project "Intelligent Vehicles and their integration in the city of the future", led by the Systems and Transportation laboratory of the University of Technology of Belfort-Montbeliard, and supported by the Regional Council of Franche Comté. The project is also a part of a program supported in the framework of Contrat Projet Etat Région (CPER). The final objective of this project is to ensure a vehicle autonomous navigation in an urban environment.

The motivation for research and development of a system that permits an autonomous navigation in urban areas results mainly from safety improvement, comfort and economic aspects. Passive safety systems such as safety belts, air bags, crumple zone or pedestrian protection systems reduce the rate of death in accidents. A further improvement is possible if active safety systems, like collision avoidance, pre-crash or active cruise control are used.

The autonomous navigation of a vehicle can be divided into three main parts. The first one is the perception of the environment. This includes determining the traversable areas like roads or lanes, obstacle detection, tracking of dynamic objects, recognition of horizontal and vertical road signs. The second part consists of localizing the vehicle in its environment. The third part is path planning of vehicle displacements, while avoiding

obstacles and collisions with dynamic and static objects.

Autonomous navigation in urban areas is, however, a long term goal. The intermediate goal is to develop Advanced Driver Assistance Systems, which are a group of systems that help a driver in its driving process.

This thesis concerns in particular the perception problem of the dynamic objects in the vehicle environment by using several sensors. This is a core part of an automatic navigation system. The goal is to detect and track dynamic objects like cars or pedestrians, and to locate them relatively to the instrumented vehicle. The estimates of objects states are an entry data of path planning and collision avoidance algorithms.

Multi-sensor based environment perception points out several research subjects: object representation, data association, and tracking. Object representation is an important part of the tracking process. It is important to find balance between precision and simplicity of the representation model. Very precise models can increase the computation complexity of the whole system since methods of the other stages should be adapted to the representation model. Data association is a crucial part of a perception system because it decides which observations originate from which objects. Wrong association leads to erroneous objects state estimation. Tracking, often called also filtering, is the process of recursive estimation of objects states having access to noisy measurements.

Our contribution concerns the object representation, data association and tracking stages. We introduce an Oriented Bounding Box (OBB) representation model and an extraction method enriched by an Inter-Rays uncertainty and a Fixed Size assumption. We propose two laser sensors fusion methods, which increase the tracking precision, especially for far objects. Concerning data association, we present a data association method based on the Nearest-Neighbour principle adapted to the OBB object representation model, which provides correct results for coalescing objects. A laser points clustering method fusing laser data with stereovision is also proposed. It allows to reliably separate ambiguous laser points clustering situations.

The development and tests of the presented work are carried out using two experimental platforms. The first one consists of a simulator that allows to generate scenarios using different 3D models. The scenarios consist of creating dynamic environments in which an instrumented vehicle evolves. The simulator allows hence to acquire data from different sensors, which are simulated with a flexible manner in terms of their position on the vehicle and their characteristics. The second experimental platform consists of an automated electrical vehicle, equipped with several sensors (cameras, laser

scanners, GPS-RTK) and communication interfaces.

The report is organized in the following manner. In chapter 2, a state of the art concerning laser points clustering, data association, object tracking, and fusion methods is presented. Chapter 3 gives the thesis statement. Chapter 4 introduces the two experimental platforms, with a detailed description of the simulator. Chapter 5 describes and evaluates an Oriented Bounding Box object representation model and a two laser scanner based tracking fusion method. This chapter presents two paradigms (Inter-Rays uncertainty and Fixed Size assumption), which are used to improve object size and position estimation. Chapter 6 presents a data association method for coalescing objects and a laser points clustering method using stereovision. Chapter 7 concludes the report and gives several perspectives and new ideas to exploit.

Chapter 2

State of the art

2.1 Laser data points clustering

In this section the problem of laser scanner data points clustering is discussed. Each scan of the scene provides a set of consecutive points (see Figure 2.1). Each points is obtained by the laser ray measurement, which has an orientation angle θ relatively to the sensor coordinate system. Each data point p_i is defined in polar coordinates by a distance and an angle (ρ_i, θ_i). The point p_i can be expressed with its Cartesian coordinates (x_i, y_i). Each laser scanner, is characterized by geometrical parameters such as the angular resolutions $\Delta\theta$ and the angular field of view. This two parameters indicate the number of measurement points that can be obtained in one scan.

The uncertainty of the measurements is described by two errors: the systematic error and statistical error. The first one does not play an important role in the process of clustering since it has the same value for all points in the scan. The second error, which is assumed to be a white noise with a normal distribution $\mathcal{N}(0, \sigma_\rho)$, is taken into consideration in clustering or line fitting algorithms. The angle coordinate θ also undergoes variations, but its standard deviation is never provided by the laser scanners manufacturers. Majority of the algorithms treating laser scanner data assumes that the angular information is not corrupted by the noise, and thus $\sigma_\theta = 0$.

The aim of clustering methods is to classify data into groups (clusters). The taxonomy and description of general clustering methods can be found in [38].

In the case of laser scanner data, however, the classical clustering meth-

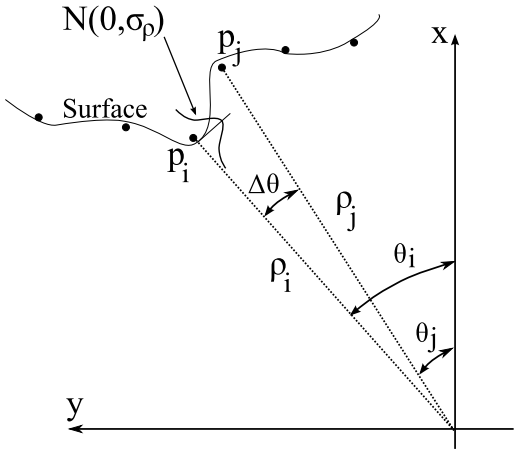

Figure 2.1: Laser scan.

ods do not give good results [40]. This is due to the special characteristics of laser scan data. The laser points do not concentrate around a cluster center, they are distributed along detected surfaces. Thus, methods, which are based on distance to center criterion, are not suitable for laser scanner data clustering. Methods that partition data into a known number of clusters cannot be used either. The number of detected objects is not provided when observing dynamic environment. In the laser data clustering, different metrics can be used. The Euclidean and Mahalanobis distances are the most popular ones.

The laser data clustering methods can be divided into two general classes: Point Distance based (PD) and Kalman Filter based (KF) [60].

2.1.1 Point Distance based methods

The Euclidean distance between two points p_i (x_i, y_i) and p_j (x_j, y_j) in the cartesian coordinate system is defined as:

$$d(p_i, p_j) = \sqrt{(x_i + x_j)^2 + (y_i + y_j)^2} \tag{2.1}$$

.

In the polar coordinate system, the Euclidian distance between p_i (ρ_i, θ_i) and p_j (ρ_j, θ_j) is expressed as:

$$d(p_i, p_j) = \sqrt{\rho_i^2 + \rho_j^2 - 2\rho_i\rho_j cos\Delta\theta} \tag{2.2}$$

where $\Delta\theta$ is the difference of the angular coordinates between the points p_i and p_j.

A simple approach consists in comparing the distance between points and a threshold d_{Th} [54, 59]. Two points p_i and p_j belong to the same cluster when $d(p_i, p_j) < d_{Th}$.

Another approach assumes that each cluster represents one surface. Thus, the distance is calculated only for consecutive points, and a cluster can be viewed as a segment. The clustering rule can be expressed as follows: if $d(p_i, p_{i+1}) < d_{Th}$, then the consecutive points p_i and p_{i+1} belong to the same segment, otherwise, two consecutive segments are considered.

There are different methods for defining the distance threshold d_{Th}. The most simple method is to set the distance threshold to a constant value. Other methods determine adaptively the distance threshold. The threshold is calculated separately of each pair of points. Using different geometrical relations between points, segments and the sensor. In [49], the following distance threshold is proposed :

$$d_{Th} = \left| \frac{\rho_i - \rho_{i+1}}{\rho_i + \rho_{i+1}} \right| \tag{2.3}$$

In this method, the threshold calculation takes into account the distance between detected surfaces and the sensor. The further the surfaces are the more difficult it becomes to separate segments.

In [21], the distance threshold is defined as follows (see Figure 2.2):

$$d_{Th} = C_0 + C_1 min(\rho_i, \rho_{i+1}) \tag{2.4}$$

where $C_1 = \sqrt{2(1 - \cos \Delta\theta)}$ and C_0 is a constant parameter that reflects the sensor noise, and which is set usually as $C_0 = 3\sigma_\rho$. σ_ρ is the sensor range uncertainty. As the former method, this technique takes into account the distance between the detected surfaces and the sensor, but it takes also into account the angular resolution, expressed by the parameter C_1.

In [65], and basing on the previous definition, the authors propose a method, which considers the surface orientation angle. An introduced parameter β allows to define a maximal surface relative to the laser rays inclination (see Figure 2.3). The distance threshold is determined as:

$$d_{Th} = C_0 + \frac{C_1 min(\rho_i, \rho_{i+1})}{\cot(\beta) \cos(\frac{\Delta\theta}{2}) - \sin(\frac{\Delta\theta}{2})} \tag{2.5}$$

The tuning of the parameter β is very important. A big value for this parameter will cause a clustering with two separate surfaces in one cluster. A small value will have the contrary effect, i.e., one surface will be detected as two separate ones. Typically, the parameter β is set to 60°.

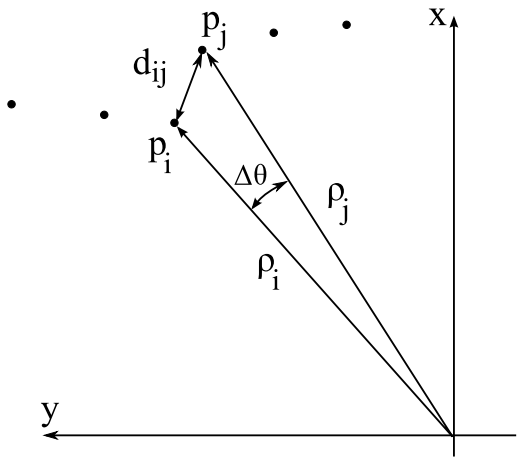

Figure 2.2: Illustration of the segmentation process presented in [21].

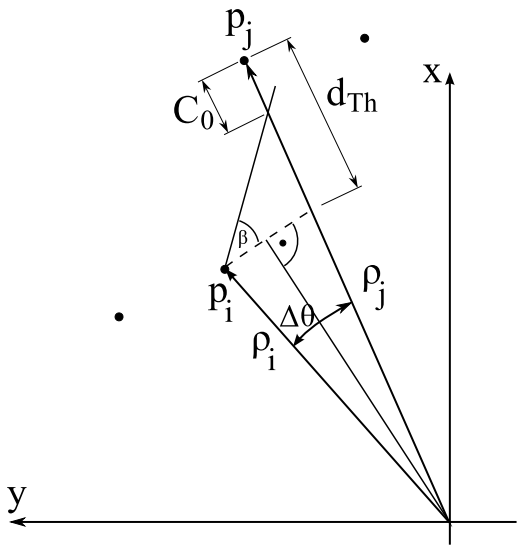

Figure 2.3: Illustration of the segmentation process presented in [65].

Anther method taking into account the surface inclination is presented in [17]. In this method, the distance threshold is expressed as:

$$d_{Th} = \rho_i \frac{\sin(\Delta\theta)}{\sin(\lambda - \theta)} + \sigma_\rho \tag{2.6}$$

where λ is the surface orientation for which the laser scanner cannot obtain correct readout due to the light reflections (see Figure 2.4). In [17], the parameter λ is set to 10°. In [40], it is proposed to set it to 25°.

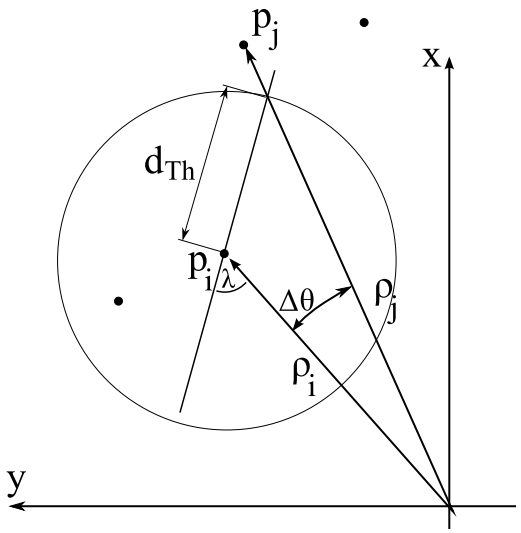

Figure 2.4: Illustration of the segmentation process presented in [17].

2.1.2 Kalman-Filter based methods

These methods detect the break point in terms of stochastic consideration. The position of the consecutive points of a segment is described by a dynamic process, governed by discrete-time stochastic difference equations:

$$x_{t+1} = F(x_t), v_t \quad (2.7)$$

$$p_{t+1} = H(x_t), w_t \quad (2.8)$$

where x is the process state, p is an observation, v is the process noise, w is the observation noise. The linearity of the functions F and H influences the type of the filter to be used. The algorithms are based on the Kalman Filter, which will be discussed in more details in section 2.3.1.

All algorithms of this group have the same form:
Repeat for $t = 1..N$, where N is the number of laser points of a segment, the following:
1. Initialise the filter $x = p_0$, $P = P_0$, where p_i is the ith point of the segment and P is covariance matrix of the state estimation.
2. Calculate the filter prediction equations
3. Do the gating for point p_t. If the point is outside the gate, extract the break point and reinitialize the filter $x = p_0$, $P = P_0$. Else, Perform the update stage and go to the step 2.
The point is considered to be inside the gate (in step 3.) if the condition $\tilde{p}_t^2 S^{-1} \tilde{p}_t \leq d_{Th}$ is met. $\tilde{p}$ is the point measurement residual, S^{-1} is

the measurement residual covariance and d_{Th} defines the threshold of the validation gate based on the sensor range uncertainty.

In [67], two Extended Kalman Filters (EKF) based approach is proposed to detect breakpoints. The authors use a non-linear dynamic system, which uses the line model expressed in the polar coordinates.

$$\phi_{t+1} = \phi_t + \Delta\theta \tag{2.9}$$

$$\rho_{t+1} = \frac{\sin(\phi_t)}{sin(\phi_t + \Delta\theta)}\rho_t \tag{2.10}$$

$$x_t = [\rho_t \quad \phi_t]^T \tag{2.11}$$

where ϕ_t is the angle between the line segment and the ray t and ρ_t is the length of the ray t, i.e., the distance between the sensor and the point p_t created by the ray t.

In this approach, the first filter, called "flexible" EKF, serves to initialize new segments, estimate the initial orientation for them and search the clutter. The second one, called "strict" EKF, allows following each initiated segment very precisely, estimating orientation of the segment and finding discontinuities.

In [17], a linear dynamic process is proposed. It is expressed by the following equations :

$$\rho_{t+1} = \rho_t + \Delta\theta\frac{d\rho_t}{d\theta} \tag{2.12}$$

$$\frac{dr_{t+1}}{d\theta} = \frac{dr_t}{d\theta} \tag{2.13}$$

$$x_t = [\rho_t \quad \frac{dr_t}{d\theta}]^T \tag{2.14}$$

Adams [2] proposes another method based on an EKF. In addition to the correct separation of planar surfaces, the proposed algorithm is suited also to detect surfaces, which "smoothly" deviate from planarity. The principle of the method is based on the calculation of a validity region, provided by the spatial gradient considering points being previously clustered as a segment.

2.1.3 Conclusion on laser data points clustering

In accordance to object representation in case of clustering, the planar surface assumption should be relaxed. Most algorithms presented in the literature assume that detected surfaces are planar. This is a common feature for

Point distance based and KF-based methods. This implies the definition of the distance threshold. The EKF-based breakpoint detection algorithm presented in [2] is not based on planar surface assumption. Nevertheless, this approach is not suitable for dynamic objects clustering. This is due to the fact that the algorithm separates correctly only surfaces, which "smoothly" deviate from planarity, what is not always the case in road environments. In addition, these approaches are not resistant to erroneous laser distance readouts such as missing or wrong distance measurement of one or more of the rays. This is due to the fact that only consecutive rays points create a single surface. By introducing 3D information from stereovision, we propose in this thesis a method that allows to relax the planar surface assumption. In addition, it is robust and provides correct clustering in the case of missing or wrong distance measurement of one or more of the rays. Indeed, the method does not force that only consecutive points can belong to the same surface.

2.2 Objects representation

2.2.1 Point based representation

Object representation is an important part of tracking systems. The object representation model expresses object characteristics that are used in the tracking process. The most basic solution is point based representation. An object state consists of position and kinematic characteristics such as velocity and acceleration. This kind of model is suitable for tracking objects for which the size is punctual from the observer point of view, or may be omitted in the tracking process. Radar based aircrafts tracking is an example of the first case [9, 10, 25, 26]. In [16] and [15], the authors use the point based representation model for road obstacle detection. The constrained environments such as highways allow introducing more assumptions about objects, and thus, the point based model representation can be used.

2.2.2 Feature based representation

The vicinity of obstacles and objects to be tracked implies more detailed object representation. In addition to the object position, the size and shape of the objects must be also taken into account. For obstacle detection and mapping, linear segments are the most popular way for representing objects in an environment [75]. They are especially used for indoor applications

[68, 5]. Line segments extracted for dynamic objects lead in most cases to represent them as rectangles. A circle representation is not suitable considering a road environment since there are not many dynamic objects having circular planar shape. In addition, a rectangular form can approximate the object shape with an acceptable accuracy.

Lines, circles and ellipses are special cases of the Conic model. The general equation for all the three representations can be formulated as:

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0 \tag{2.15}$$

By considering $a = b = c = 0$, the line equation is obtained. The circle equation is obtained with $a = c$ and $b = 0$. The ellipse equation is defined when $(b^2 - ac) < 0$.

2.2.3 Rectangle based representation

Rectangle based representation model is most common for dynamic objects tracking. It provides good approximation of real object size, especially for road vehicles since their planar shape is generally rectangular. Another important aspect of this representation model is its compactness. Indeed, the model has a small number of parameters characterizing the representation. Most of the existing approaches of rectangle extraction are based on line fitting algorithm. The first step of the rectangle extraction algorithm is to fit line segments to laser data points. This stage can be prefaced by a clustering algorithm, but it is not always necessary.

Line fitting algorithm

The line fitting algorithm is used to approximate the original run of the scan points. As a result, parameters defining lines and their co-variances are obtained. The cartesian form of a line equation is defined as follows:

$$ax + by + c = 0 \tag{2.16}$$

The equivalent polar form is given as follows:

$$\rho \cos(\theta - \alpha) - r = 0 \tag{2.17}$$

where α and r are the parameters of the line model.

One of the most popular methods used to fit the mathematical model of a line to data is linear regression. This method solves the minimum error

along the vertical axis. The method used to fit lines to laser data points is presented in [75, 68].

In [5], a weighted total-linear regression method is proposed. The total-line regression minimizes the mean square error calculated as a distance from the points to the line. The method finds the line equation and covariance matrix in the polar coordinates system. Data points are described by their position and range uncertainty. The angular uncertainty is neglected. The point weights w express the points uncertainty. The solution is expressed as follows :

$$\tan(2\alpha) = \frac{\frac{2}{\sum w_i}\sum_{i<}\sum_{j} w_i w_j \rho_i \rho_j \sin(\theta_i + \theta_j) + \frac{1}{\sum w_i}\sum (w_i - \sum w_j)\rho_i^2 \sin 2\theta_i}{\frac{2}{\sum w_i}\sum_{i<}\sum_{j} w_i w_j \rho_i \rho_j \cos(\theta_i + \theta_j) + \frac{1}{\sum (w_i} - \sum w_j) w_i \rho_i^2 \cos 2\theta_i} \tag{2.18}$$

$$r = \frac{\sum w_i \rho_i cos(\theta_i - \alpha)}{\sum w_i} \tag{2.19}$$

Less computationally complex, the equivalent cartesian form is defined as follows:

$$\tan(2\alpha) = \frac{-2\sum w_i(\bar{y}_w - y_i)(\bar{x}_w - x_i)}{\sum w_i[(\bar{y}_w - y_i)^2 - (\bar{x}_w - x_i)^2]} \tag{2.20}$$

$$r = \bar{x}_w \cos\alpha + \bar{y}_w \sin\alpha \tag{2.21}$$

where $\bar{x}_w = (\sum w_i \sum w_i \rho_i \cos\theta_i)^{-}1$ and $\bar{y}_w = (\sum w_i \sum w_i \rho_i \sin\theta_i)^{-}1$ are weighted means.

The co-variance equations are:

$$\sigma_\alpha^2 = \frac{1}{(D^2 + N^2)^2}\sum w_i^2 (A - B)^2 \sigma_{\rho_i}^2 \tag{2.22}$$

where $A = N(\bar{x}_w \cos\theta_i - \bar{y}_w \sin\theta_i - \rho_i \cos 2\theta_i)$
and $B = D(\bar{x}_w \sin\theta_i + \bar{y}_w \cos\theta_i - \rho_i \sin 2\theta_i)$. D and N are the denominator and the numerator of the right hand side of the equation (5.8), respectively.

$$\sigma_r^2 = \sum [\frac{w_i}{\sum w_j} cos(\theta_i - \alpha) + \frac{\partial\alpha}{\partial\rho_i}(\bar{y}_w \cos\alpha - \bar{x}_w \sin\alpha)]^2 \sigma_{\rho_i}^2 \tag{2.23}$$

$$\sigma_{\alpha r} = \sum \frac{\partial\alpha \partial r}{\partial P_i \partial P_i}\sigma_{\rho_i}^2 \tag{2.24}$$

where $P_i = (\rho_i, \theta_i)$ is a point measurement.

The detailed equations with derivation can be found in [4].

In [24], Duda proposes an Iterative End-Point Fit method (IEPF), which is a recursive algorithm. It starts by finding the line which crosses the first and the last points in a given cluster. In the next step, the point with the maximum distance to the line is considered to test the following rule. If the distance is greater that a certain threshold, then the point divides the line segment into two clusters, and the first step is performed for each of the clusters. To obtain the line approximation, the line fitting algorithm has to be executed for separated segments.

Other authors use the Split-and-Merge (SM) algorithm [57], which is very similar to the IEPF method, described above. The difference concerns the line creation. Instead of taking only the first and last points, the line fitting algorithm is performed for all points. The SM method is very popular and is the fastest compared with the other methods (see [56]). Furthermore, the method returns line approximations.

In [5], the authors present a line fitting algorithm, which is based on Hough Transform [37]. All measurement points are transformed into a line parameter domain (ρ, θ). This leads to a clustering process of n points. The center of each cluster defines one line segment. The consecutive configuration of laser data points for each surface allows reducing the computation complexity. The drawback of this method is its implementation complexity, and the fact that the line fitting algorithm has to be performed for line approximation.

Another approach is based on an incremental algorithm, which is called Line-Tracking [72]. The algorithm starts by constructing a line passing by the two first points. Then, a new point is added to the current line model. In the next step, the line parameters are recomputed. If the parameters satisfy the line condition, the next point is considered and the algorithm continues. Otherwise, the line is returned and the algorithm starts with the next two points. Simplicity is the advantage of the method, as well as the fact that the line approximation is performed parallelly. This means that the method provides directly fitted lines.

Oriented rectangle based representation

In [74], an oriented rectangle model is used to represent dynamic objects. Extraction of rectangles consists of two stages: modified least-square fit and corner fitting. The authors propose refinements to the least-square fit

method to reduce problems with objects having rounded corners. The high point density of rounded corners causes the line to rotate away from more distant points. The modified least-square fit algorithm consists of three steps. The first one performs an unweighted line fitting for all points in the segment and sets the weights of the points basing on their separation along the line. The second step achieves a trial weight fit using 80% of the points having the best unweighted fit. This allows detecting outlier points. The third step consists of the final weight fit.

The fitted line serves to corner fitting. The points are divided into two sets. The division is defined by the "knuckle" point, which is the farthest point from the fitted line. The geometrically longer part is used to determine the object direction by line fitting. Next, the position of the shorter line is found basing on the assumption that there is right angle between the two sides.It is done by taking the mean position along the longer side. In the last stage, the goodness of the fit is tested. This is achieved by doing unconstrained linear fit on the short side and checking the obtained angle between the two sides. The angle must not differ more that 50 degrees from the right angle to be considered as a good fit.

In [70], an Oriented Bound Box (OBB) object representation is proposed. The first step of the proposed algorithm consists of the vectorization of data points in a given cluster. The principal idea of this operation is similar to the IEPF method, described in section 2.2.3. The vectorization process is iterative and starts by describing the vector by the first and the last points in the cluster. The vector is subdivided until the maximum distance between every scan point and the current vector is smaller than a given threshold. This step is repeated for every resulting vector. The longest vector indicates the orientation angle of the OBB. After the extraction of the OBB, the quality of the extracted primitive is computed. The quality is expressed by the mean and divergency of the distance of the scan point to the closest side of the box. The advantage of the method is its low computational complexity.

An oriented rectangle representation is proposed in [59]. The tracking system is based on Particle Filter (PF). In the PF framework, the observation stage of the filtering consists of computing the likelihood for the measurement and each of the particles. The separate points are not used to create a measurement primitive. They are used to compute the likelihood of the laser points set given the predicted particle. The measurement likelihood is a sum of all points' likelihood. The point likelihood is a function of the rays range uncertainty and a cost, which depends on the rectangle relative

point position. The advantage of this approach is that one algorithm is used for representation, data association and tracking. Its drawback is that two targets may in principle merge into one. Rao-Blackwellized PF is used in order to reduce the number of parameters being estimated by Monte Carlo sampling [59].Nevertheless, this solution is more computationally complex than KF based methods.It is shown that PF gives better accuracy than the other methods, but the accuracy is positively related with the number of particles.

2.2.4 Conclusion on object representation

The point based object representation model can be used with success in tracking of distant objects. It is possible to use this model since the size of the tracked objects is small relative to the distance. This model, however, is not precise enough for applications in road environments, except from well structured environments such as highways, where the objects maneuvers and positions are limited. In such environments, when taking into consideration autonomous cruise control system (ACC), the representation model with limited geometrical form precision such as point, ellipse or Axis Aligned Bounding Box (AABB) may be acceptable. Nonetheless, in constrained environments like urban road, where there are less constraints on objects movements, position and orientation, more precision in terms of object geometry approximation models is necessary. In the literature, the oriented rectangles are used [65]. The drawback of the existing methods is that the rectangle representing an object is obtained by line and corner fitting. This approach assumes that the objects are constructed by planar surfaces. In [74], a modified line fitting algorithm is proposed so it can cope with rounded corners. It is achieved by omitting some points. The number of points to be omitted is supposed constant during the tracking. This hypothesis does not allow giving always satisfactory results for all kinds of objects. The OBB based representation model is proposed in [70]. The drawback of the method is that it is based on a planar surface assumption. The orientation angle found by taking the longest extracted vector, representing the surface, may cause also a problem. The theorem proved in [27] says: The rectangle of minimum area enclosing a convex polygon has a side collinear with one of the edges of the polygon. Thus, taking the longest side of the convex polygon does not guarantee the optimal OBB fit. Figure 2.5 presents the example where taking the longest side of the convex polygon results in bad OBB alignement.

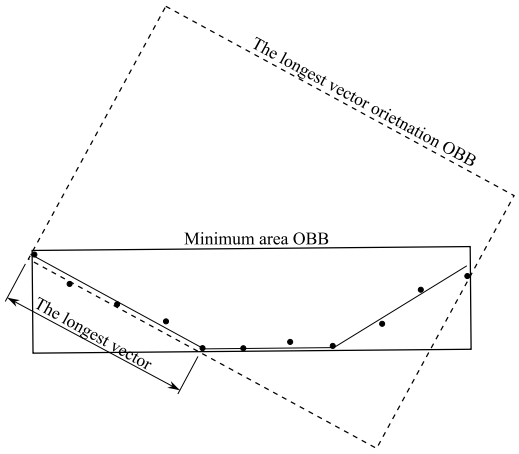

Figure 2.5: Comparison of the longest vector orientation OBB with minimum area OBB.

In the literature, there exists no approach, which takes into account the fact that the extreme points of the detected object do not represent the real object edges. Thus, Inter-Rays uncertainty is proposed in this thesis to introduce a parameter, which is used in size estimation and uncertainty computation.

2.3 Tracking

The aim of the object tracking is estimation of the current object state using noisy measurement of the objects state. The process of state transitions is a stochastic process having Markov Chain characteristics. It means that all state transitions are probabilistic and a state in time t depends only on a state in time $t-1$.

$$p(x_t|x_1, x_2, ..., x_{t-1}) = p(x_t|x_{t-1}) \tag{2.25}$$

The first two stochastic filters were presented independently by Wiener [79, 78] and Kolmogorov [47]. The first one, is continuous-time filter and the second one is discrete-time filter. Both solutions are not recursive, and thus, cannot be used in objects tracking.

Kalman filter (KF) is the most popular recursive filter. Its applicability is limited, however, to linear dynamic systems with Gaussian noise. The modification of KF like Extended Kalman Filter (EKF) or Unscented Kalman Filter (UKF) can be used in case of non-linear systems. The sequen-

tial Monte Carlo method allows to use other types of probability functions to describe the uncertainty.

2.3.1 Kalman Filter

The Kalman Filter [43] is an optimal, in terms of Minimum Mean Square Error (MMSE) criterion, recursive filter for a linear dynamic system driven by white noise. The discrete-time linear dynamic system is described by the two equations: dynamic equation and measurement equation. The first one describes the model of state transition at time t to $t+1$:

$$x_{t+1} = F_t x_t + C_t u_t + G_t v_t \tag{2.26}$$

F_t is a state transition matrix. C_t is an input matrix applied to the known input vector u_t. v_t is a Gaussian zero-mean process noise with a covariance matrix Q_t and G_t is the transition matrix of the process noise v_t.

The measurement equation is expressed as follows:

$$z_t = H_t x_t + w_t \tag{2.27}$$

where z_t is an observation of the state x_t at time t, H_t is an observation matrix and w_t is a white measurement noise with a covariance matrix R_t.

The two noise sequences are assumed mutually independent.

$$E\left[v_i v_j^T\right] = \delta_{i,j} Q_i \tag{2.28}$$
$$E\left[w_i w_j^T\right] = \delta_{i,j} R_i \tag{2.29}$$

At time $t = 0$,

$$x_0 \approx \mathcal{N}(\hat{x}_0, P_{0|0}) \tag{2.30}$$

The filtering algorithm consists of two stages: prediction and update. Prediction stage uses the previous iteration state estimation to produce *a priori* state estimation at the current time. *A priori* state estimation does not include information from measurements. In the update stage, the state estimation is improved by incorporating measurements information, and *a posteriori* estimation is obtained. The update state formula is governed by a Kalman gain. The Kalman gain is calculated basing on the error covariance matrices of the *a priori* estimation and measurement.

The algorithm equations are expressed as follows:

$$p(x_{t-1}|z_{1:t-1}) = \mathcal{N}(x_{t-1}; \hat{x}_{t-1|t-1}, P_{t-1|t-1}) \quad (2.31)$$
$$p(x_t|z_{1:t-1}) = \mathcal{N}(x_t; \hat{x}_{t|t-1}, P_{t|t-1}) \quad (2.32)$$
$$p(x_t|z_{1:t}) = \mathcal{N}(x_t; \hat{x}_{t|t}, P_{t|t}) \quad (2.33)$$

with :

$$\hat{x}_{t|t-1} = F_{t-1}\hat{x}_{t-1|t-1} + C_{t-1}u_{t-1} \quad (2.34)$$
$$P_{t|t-1} = F_{t-1}P_{t-1|t-1}F_{t-1}^T + G_{t-1}Q_{t-1}G_{t-1}^T \quad (2.35)$$

$\hat{x}_{t|t-1}$ and $P_{t|t-1}$ are respectively the state prediction and the covariance matrix of the prediction error.

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t\nu_t \quad (2.36)$$
$$P_{t|t} = P_{t|t-1} - K_tS_tK_t^T \quad (2.37)$$

$\hat{x}_{t|t}$ and $P_{t|t}$ are respectively the state estimation and the covariance matrix of the estimation error. K_t is the Kalman gain, defined at time t as follows:

$$K_t = P_{t|t-1}H_t^T\left[H_tP_{t|t-1}H_t^T + R_t\right]^{-1} \quad (2.38)$$

ν_t is the innovation, i.e. the difference between the measurement and the prediction, given by:

$$\nu_t = z_t - H_t\hat{x}_{t|t-1} \quad (2.39)$$

and the covariance matrix is:

$$S_t = H_tP_{t|t-1}H_t^T + R_t \quad (2.40)$$

In the case where the transition function or measurement function are non-linear or if the noises v_t or w_t are not gaussian, determinist non optimal filters, based on a linear Kalman filter algorithm, are usually used. The Extended Kalman Filter (EKF) is based on a local linearisation of the non-linear transition and measurement functions. The Unscented Kalman Filter (UKF) uses a deterministic sampling technique known as the unscented transform (UT) to pick a minimal set of sample points (called sigma points) around the mean. These sigma points are then propagated through the non-linear functions, from which the mean and covariance of the estimate are then recovered. The result is a filter, which captures more accurately the true mean and covariance.

The two following sections describe the EKF and the UKF.

2.3.2 Extended Kalman Filter (EKF)

Let us denote the non-linear transition and observation models by:

$$x_{t+1} = f_t(x_t, v_t) \tag{2.41}$$
$$z_t = h_t(x_t) + w_t \tag{2.42}$$

with v_t and w_t are respectively the process and measurement noises.

The function f can be used to compute the predicted state from the previous estimate and similarly the function h can be used to compute the predicted measurement from the predicted state. However, f and h cannot be applied to the covariance directly. Instead a matrix of partial derivatives (the Jacobian) is computed.

At each time-step the Jacobian is evaluated with current predicted states. These matrices can be used in the Kalman filter equations. This process essentially linearizes the non-linear function around the current estimate.

The EKF makes the hypothesis that the *a posteriori* probability density function $p(x_t|z_{1:t})$, that is not gaussian, can be approximated by a Gaussian density with the following recurrences:

$$p(x_{t-1}|z_{1:t-1}) \approx \mathcal{N}(x_{t-1}; \hat{x}_{t-1|t-1}, P_{t-1|t-1}) \tag{2.43}$$
$$p(x_t|z_{1:t-1}) \approx \mathcal{N}(x_t; \hat{x}_{t|t-1}, P_{t|t-1}) \tag{2.44}$$
$$p(x_t|z_{1:t}) \approx \mathcal{N}(x_t; \hat{x}_{t|t}, P_{t|t}) \tag{2.45}$$

with:

$$\hat{x}_{t|t-1} = f_{t-1}\hat{x}_{t-1|t-1} \tag{2.46}$$
$$P_{t|t-1} = \hat{F}_{t-1}P_{t-1|t-1}\hat{F}_{t-1}^T + G_{t-1}Q_{t-1}G_{t-1}^T \tag{2.47}$$
$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t\left[z_t - h_t(\hat{x}_{t|t-1})\right] \tag{2.48}$$
$$P_{t|t} = P_{t|t-1} - K_t S_t K_t^T \tag{2.49}$$
$$K_t = P_{t|t-1}H_t^T\left[H_t P_{t|t-1}H_t^T + R_t\right]^{-1} \tag{2.50}$$

$\hat{F}_t$ and $\hat{H}_t$ are the local linearisation of the functions f_t and h_t respectively around the estimate and the prediction:

$$\hat{F}_t = \left.\frac{\partial f_t}{dx_t}\right|_{x_t=x_{t|t}} \tag{2.51}$$
$$\hat{H}_t = \left.\frac{\partial h_t}{dx_t}\right|_{x_t=\hat{x}_{t|t}} \tag{2.52}$$

2.3.3 Unscented Kalman Filter (UKF)

The Unscented Kalman Filter, proposed in [41], approximates an *a posteriori* probability density by a Gaussian function as for EKF. But instead of approximating the non-linear transition and measurement functions, UKF approximates the probability density by a set of weighted points that are chosen deterministically. These points are transformed by the transition and measurement non-linear functions to obtain the new probability density. This approximation is called the Unscented Transform.

Unscented Transform

The unscented transform is a method that permits to calculate the statistics of a random variable transformed by a non-linear function notation [76, 42].

Let us consider a non-linear system, defined as:

$$y = f(x) \tag{2.53}$$

with x is a random variable with mean $\overline{x}$ and covariance P_{xx}, and y is a random variable whose statistics are to be determined.

A set of points is choosen deterministically so that the mean and the covariance are respectively $\overline{x}$ and P_{xx}. The non-linear function f is applied on each point to obtain a set of transformed points of mean $\overline{y}$ and covariance P_{yy}.

The probability density of a random variable x of dimension n, with mean $\overline{x}$ and covariance P_{xx}, is approximated by $2n+1$ weighted points, given by:

$$\begin{array}{ll} \mathcal{X}_0 = \overline{x} & W_0 = \frac{\kappa}{n+\kappa} \\ \mathcal{X}_i = \overline{x} + \left(\sqrt{(n+\kappa)\,P_{xx}}\right)_i & W_i = \frac{1}{2(n+\kappa)} \\ \mathcal{X}_{i+n} = \overline{x} - \left(\sqrt{(n+\kappa)\,P_{xx}}\right)_i & W_{i+n} = \frac{1}{2(n+\kappa)} \end{array} \tag{2.54}$$

where $\kappa \in \Re$, $\left(\sqrt{(n+\kappa)\,P_{xx}}\right)_i$ is the *i-th* line or column of the square root matrix of $(n+\kappa)\,P_{xx}$ and W_i is the weight associated to the *i-th* point.

The procedure of the transformation is as follows:

1. Transform each point $\mathcal{X}_i$ by the non-linear function f to obtain the set of transformed points:

$$\mathcal{Y}_i = f(\mathcal{X}_i) \tag{2.55}$$

2. The mean $\overline{y}$ is given by the weighted mean of the transformed points

$$\overline{y} = \sum_{i=0}^{2n} W_i \mathcal{Y}_i \tag{2.56}$$

3. The covariance matrix P_{yy} is given by :

$$P_{yy} = \sum_{i=0}^{2n} W_i (\mathcal{Y}_i - \overline{y})(\mathcal{Y}_i - \overline{y})^T \tag{2.57}$$

Unscented Kalman Filter Algorithm

Let us consider the transition and measurement models, given by:

$$x_{t+1} = f_t(x_t, v_t) \tag{2.58}$$
$$z_t = h_t(x_t) + w_t \tag{2.59}$$

where v_t and w_t are respectively the noises on the transition and measurement processes, that are supposed to be mutually independent and zero mean Gaussian white noise. Their covariance matrices are respectively:

$$E\left[v_i v_j^T\right] = \delta_{i,j} Q_i \tag{2.60}$$
$$E\left[w_i w_j^T\right] = \delta_{i,j} R_i \tag{2.61}$$

It is supposed that:

$$E\left[v_i v_j^T\right] = 0 \quad \forall i, j \tag{2.62}$$

Let us note:

$$\mathcal{X}^a = \left[(\mathcal{X}^x)^T (\mathcal{X}^v)^T\right] \tag{2.63}$$

and

$$n = \dim(x_t) + \dim(v_t) \tag{2.64}$$

The UKF algorithm is as follows:

1. **Computation of the points of approximation**

$$\mathcal{X}^a_{t-1} = \begin{bmatrix} \hat{x}^a_{t-1} & \hat{x}^a_{t-1} + \sqrt{(n+\kappa)P^a_{t-1}} & \hat{x}^a_{t-1} - \sqrt{(n+\kappa)P^a_{t-1}} \end{bmatrix} \tag{2.65}$$

with

$$\hat{x}^a_{t-1} = \begin{bmatrix} \hat{x}^T_{t-1|t-1} & 0^T_{dim(v_{t-1})} \end{bmatrix}^T \tag{2.66}$$

and

$$P^a_{t-1} = \begin{pmatrix} P_{t-1|t-1} & 0 \\ 0 & Q_{t-1} \end{pmatrix} \tag{2.67}$$

2. **Computation of the associated weights**
   To compute the associated weights, use the equation (2.54).

3. **Prediction**

$$\mathcal{X}^x_{t|t-1} = f_{t-1}(\mathcal{X}^x_{t-1}, \mathcal{X}^v_{t-1}) \tag{2.68}$$

$$\hat{x}_{t|t-1} = \sum_{i=0}^{2n} W_i \mathcal{X}^x_{i,t|t-1} \tag{2.69}$$

$$P_{t|t-1} = \sum_{i=0}^{2n} W_i (\mathcal{X}^x_{i,t|t-1} - \hat{x}_{t|t-1})(\mathcal{X}^x_{i,t|t-1} - \hat{x}_{t|t-1})^T \tag{2.70}$$

$$\mathcal{Z}_{t|t-1} = h_t(\mathcal{X}^x_{t|t-1}) \tag{2.71}$$

$$\hat{z}_{t|t-1} = \sum_{i=0}^{2n} W_i \mathcal{Z}^x_{i,t|t-1} \tag{2.72}$$

4. **Estimation**

$$S_t = P_{v_t v_t} = R_t + \sum_{i=0}^{2n} W_i (\mathcal{Z}_{i,t|t-1} - \hat{z}_{t|t-1})(\mathcal{Z}_{i,t|t-1} - \hat{z}_{t|t-1})^T \tag{2.73}$$

$$P_{x_t z_t} = \sum_{i=0}^{2n} W_i (\mathcal{X}^x_{i,t|t-1} - \hat{x}_{t|t-1})(\mathcal{Z}^x_{i,t|t-1} - \hat{z}_{t|t-1})^T \tag{2.74}$$

$$K_t = P_{x_t z_t} S_t^{-1} \tag{2.75}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(z_t - \hat{z}_{t|t-1}) \tag{2.76}$$

$$P_{t|t} = P_{t|t-1} - K_t S_t K_t \tag{2.77}$$

2.3.4 Particle Filter

Bayesian filtering provides a convenient framework for objects tracking due to the weak assumptions on the state space model and the first order Markov chain recursive properties. Monte Carlo methods and more specifically particle filters, based on Bayesian inference, have been extensively employed for tracking problems [18, 31, 61]. Multi-modality, in particular, enables the system to evolve in time with several hypotheses on the state in parallel. This property is practical to corroborate or reject an eventual track after several frames. Particle filters rely on Sequential Monte Carlo (SMC) methods. A large number of samples $x_k^i, i = 1...N_S$ are drawn from the posterior distribution $p(x_k|z_k)$. It follows from the law of large numbers that:

$$p(x_k|z_k) \approx \sum_{i=1}^{N_S} w_k^i \delta(x_k - x_k^i) \tag{2.78}$$

where w_k^i are the weights, i.e. $\sum w_k^i = 1$, and $\delta(.)$ is the Kronecker delta function. However, because it is often difficult to draw samples from the posterior probability density function (pdf), an importance density $q(.)$ is used to generate the samples x_k^i. It can be shown that [6]:

$$w_k^i \approx w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)} \tag{2.79}$$

The choice of the importance density is crucial to obtain a good estimate of the posterior pdf. It has been shown that the set of particles and associated weights $\{x_k^i, w_k^i\}$ will eventually degenerate, *i.e.* most of the weights will be carried by a small number of samples and a large number of samples, will have negligible weight [69]. In such a case, and because the samples are not drawn from the true posterior pdf, the degeneracy problem cannot be avoided, and resampling of the set needs to be performed. Nevertheless, the closer the importance density is from the true posterior density, the slower the set $\{x_k^i, w_k^i\}$ will degenerate; a good choice of importance density reduces the need for resampling.

2.4 Data association

Data association is a crucial part of all objects tracking systems. This stage allows to correlate observation data obtained in a current time-step

with existing objects thanks to their estimation from the previous time-step. False data association leads to erroneous object state estimation. The association algorithm serves to extract object measurement from the clutter and to separate multiple objects.

Each tracked object state evolves according to a dynamic model used in the filter. Thus, the prediction stage provides information, which can be used to associate the observation with the tracks. *A priori* state estimate and its covariance allow to define a validation region for each track. This allows to reject measurements which cannot originate form the object. This mechanism is not enough since very often, in a track validation gate, there are more than one measurement. This is due to the clutter being the result of the sensor noise or to the presence of other objects measurements. The second case takes place when the validation gates of different objects intersect.

2.4.1 Nearest-Neighbour

The Nearest-Neighbour (NN) method associates the closest measurement from all present in the validation gate of the track [9, 14]. Mahalanobis distance d_{ij}^2 is used to calculate the distance between a measurement $z_i(t)$ and the track measurement prediction $\hat{z}_j(t)$. It is expressed as follows:

$$\gamma \leq d_{ij}^2 = \tilde{z}_{ij}(t)^T S_i^{-1} \tilde{z}_{ij}(t) \tag{2.80}$$

where γ is a gate threshold, found from χ^2 distribution, by using the confidence interval of a n states of freedom (related to the dimension of the measurement domain). $\tilde{z}_{ij}$ is the Euclidean distance between the track predicted measurement $\hat{z}_i$ and the measurement $\hat{z}_j$. S is the innovation matrix defined as:

$$S_i^t = H_i(t) P_i(t|t-1) H_i(t)^T + R(t) \tag{2.81}$$

The use of the innovation matrix allows to take into account the noise of the transition dynamic system $H_i(t)$ and measurement system $R(t)$. The measurement uncertainty is related to the sensor noise. The method is not adapted to problems like existence of clutter in the sensory data or no object detection. In this conditions, the NN method can give false association results.

2.4.2 PDA

The Probabilistic Data Association is suited for single track case. Instead of considering only one measurement among the received ones and discard the rest, the PDA method takes all validate measurements by using weighted a mean [11]. The weights are calculated in real-time and reflect the measurement origin uncertainty. The uncertainty is represented by the probability obtained in the process of Bayesian inference. The PDA principles are applicable to the task of data association in the form of recursive state estimator (tracker), called PDA Filter (PDAF). The PDAF is based on the KF scheme. The algorithm assumes that there is only one track, which is already initialized. The target uncertainty is represented by a normal probability density function.

The process of association, executed in each filter iteration, starts by creating a validation gate as in the case of the NN algorithm (equation 2.80). In this approach, the association events are mutually exclusive and exhaustive for multiple validated measurements. Each event h_i of m ones represents the fact that measurement z_i is a target originated, where m is the number of validated measurements. There is also an additional event h_0, which represents the case where none of the measurements is a target originated. Basing on the total probability theorem with regard to the association event, a conditional mean of the state at time t is defined:

$$\hat{x}(t|t) = \sum_{i=0}^{m(t)} \hat{x}_i(t|t)\beta_i(t) \tag{2.82}$$

where $\hat{x}_i(t|t)$ is the updated state conditioned on the event that validated the measurement z_i is correct and $\beta_i(t)$ is the conditional probability of that event.

The conditional mean serves to provide the state estimation update equation is defined as follows :

$$\hat{x}(t|t) = \hat{x}(t|t-1)W(k)\tilde{z}(t) \tag{2.83}$$

The combined innovation $\tilde{z}(k)$ is a expressed as follows:

$$\tilde{z}(t) = \sum_{i=0}^{m(t)} \beta_i(t)\tilde{z}_i(t) \tag{2.84}$$

having residuals $\tilde{z}_i(t) = z_i(t) - \hat{z}_i(t)$.

The events probabilities are obtained by a normalization:

$$\beta_i(t) = \frac{\beta_i^{'}(t)}{\sum_{i=0}^{m} \beta_i^{'}(t)} \tag{2.85}$$

where the H_0 false detection event probability is given by:

$$\beta_0^{'}(t) = p_F^m(1 - p_D) \tag{2.86}$$

where p_F is the probability density of the false measurement, and p_D is the probability of the target detection. The other event probabilities are expressed as follows:

$$\beta_i^{'}(t) = \frac{p_F^{m-1} p_D e^{-\frac{d_{ij}^2(t)}{2}}}{(2\pi)^{M/2}\sqrt{|S_i(t)|}} \tag{2.87}$$

where S_i, d_{ij}^2 are the same as in equations (2.80) and (2.81).

The rest of the algorithm consists of using a standard Kalman Filter equation with the modified update equation of covariance P. This modification is intended to reflect the effect of uncertain correlation. For details, one can refer to [11].

2.4.3 MHT

The Multi Hypothesis Tracking, first proposed in [62], is an approach which can be used in the presence of multiple tracks. In addition to measurements assigned to the existing tracks, it supports also track initialization [9]. The principle of this approach consists in taking into account the most probable hypotheses, where each new observation produces a new hypothesis. For a new observation, three hypotheses are created. They express the following cases: an observation is a false alarm, an observation is originating from an existing track, an observation originates from a new object. The evaluation of the association probabilities are computed by using track state estimation with its covariance matrix and probability of associating a new observation with an existing track. It is assumed that probability density functions are Gaussians.

The constructed hypotheses create a tree graph. Edges of the graph represent the hypothesis that a measurement i originates from a track j. Thus, for each observation, at least, two edges are created (association to an existing track and a new track initialization) (see Figure 2.6)*. In the figure 2.6 a square represents a measurement and a circle represents a track.

*The figure is extracted from [62].

Each line represents a possible associations of the measurement with the tracks. Number of leaves (the bottom graph nodes) gives the number of possible configurations to be tested.

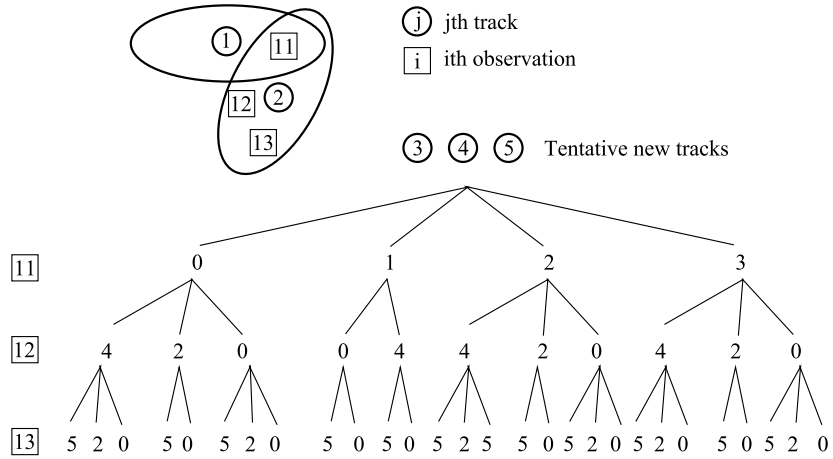

Figure 2.6: Example of hypothesis tree graph for a given configuration of targets and measurement. The figure is extracted from [62].

The computation complexity of this approach is exponential. Thus existence of many tracks and measurements implies long time calculation of the algorithm. To reduce the complexity the regrouping of hypothesis is proposed in [62]. The idea is to reduce the number of probabilities to be calculated by treating separately the hypotheses which do not influence a certain group of hypotheses. Another possibility is to reject the hypothesis with low probability.

2.4.4 JPDA

The Joint Probabilistic Data Association is the evolution of PDA approach to multi-track association, and can be seen as a particular case of MHT approach which minimizes the number of hypotheses to be checked. This is obtained by regrouping tracks into groups. Each such a group contains the tracks with overlapping validation gates. The tracks from one group share the same measurements. Each group is treated separately, and thus the complexity can be reduced. The joint probability of association will be calculated only for the tracks in the same group, what reduces the dimensions of JPDA to the number of the tracks in the group. After first step of tracks grouping, the hypothesis creation calculation takes place. Every hypothesis represents a particular way the measurements can be associated with tracks

from one group. It is assumed that each track generates one measurement, which may not be detected, and each measurement corresponds to only one track. The time reference (t) is omitted for simplicity. Next, probabilities for each hypothesis are calculated by:

$$p'(h_l) = p_F^{(N_m - N_t + N_{nd})}(1 - p_D)^{N_{nd}} p_D^{(N_t - N_{nd})} g_{ij} \cdots g_{mn} \tag{2.88}$$

where p_F is the probability density for false returns. p_D and $(1 - p_D)$ are the probabilities of target detection and target missing, respectively. N_m is the total number of measurements, N_t is the total number of targets, N_{nd} is the number of not detected targets. g_{ij} is the probability density that the measurement i originates from the target j. It is expressed as follows:

$$g_{ij} = \frac{e^{-\frac{d_{ij}^2}{2}}}{(2\pi)^{M/2}\sqrt{|S_i|}} \tag{2.89}$$

The number of multipliers g_{ij} is equal to $N_t - N_{nd}$.

The normalization finishes the computation of hypotheses probabilities, similar as in the case of the PDA:

$$p(h_l) = \frac{p^{'}(h_l)}{\sum_{l=0}^{N_h} p_{(}^{'} h_l)} \tag{2.90}$$

where N_h is the total number of hypotheses.

The last step of the algorithm consists of computing the probability of associating the ith measurement with the jth track. It is obtained by calculating the sum of the probabilities of the hypotheses in which the event occurs.

$$\beta_i(t) = \sum_{l \in L_i} p(h_l) \tag{2.91}$$

where L_i is a set of numbers of hypotheses, which include the event that the ith measurement originates from the jth track.

After, the algorithm continues with the computation of the combined innovations (equation 2.84) of the standard PDA.

2.4.5 Conclusion on data association techniques

The presented standard mulit-track association methods are not suitable to the task of road environment tracking without modifications. The basic assumption in the probabilistic methods is that only one measurement originates from a track. If we treat the raw data points as measurement

and consider an OBB track model, it is clear that this assumption is broken. Even if this assumption was relaxed, the number of hypotheses to be computed in the case of two close objects would be prohibitive for real-time algorithm execution. Indeed, there are many laser data points treated as measurements, even in presence one or two tracks. One can try to associate the features (eg. line segments) to the tracks. However, the problem arises when the sides of two close positioned objects are co-linear. The points originating form the two surfaces would be clustered into one segment. In the thesis the NN based association adapter to the OBB object representation model is proposed. The motivation for using NN principle was low computational complexity. The size information of the object is used to increase the robustness of not flexible NN approach.

2.5 Data fusion

The aim of data fusion is to combine data from many independent sources in order to produce information more accurate and reliable than when using each input source separately. Input data may not only originate from different sensors, but also from the same sensor at different moments. It is also possible to use different experts conclusions made for the same data. The advantage of sensor fusion over one sensor processing consists in redundancy, diversity and complementarity between multiple sensors:

- One can talk about redundancy when there are multiple sensors measuring the same entity. The measured quantity is often correlated, while the uncertainty of used sensors is usually uncorrelated. Thus, the sensors redundancy allows reducing the uncertainty.

- Sensors of different nature, which measure the same entity using different technologies, introduce sensor diversity. The spatial and time diversity is obtained when sensors measure the same scene, but from different locations and in different moments.

- Sensors observing subsets of the environment space provide complementary data. By the union of the subsets, broader view of the environment may be obtained.

The fusion process can be categorized by the processing stage at which fusion takes place:

Low level fusion or data fusion integrates different sources raw data to obtain data of the same form as an original signal, which is more synthetic and contains more information.

Intermediate level fusion or feature level fusion combines various features, extracted from different sources raw data or from the same raw data. The objective is to obtain a uniform overall feature map.

High level fusion or decision fusion combines decisions or scores coming from several experts. Hard fusion concerns combination of decisions while soft fusion concerns score combination.

Fusion algorithms can be classified, depending on methods they use [51], into four groups: estimation methods, inference methods, artificial intelligence methods and classification methods. The two first groups of methods are widely used in the context of road environment perception systems.

Estimation methods provide a fused value by taking the weighted average of redundant information. These kinds of methods are used in real-time processing of dynamic low-level data. These methods can be regrouped into two groups: non-recursive and recursive methods. In the first group, there are weighted average based and least squares based methods. The methods of the second group consist in general of filters such as Kalman Filter and Particle Filter. The most popular methods are based on Kalman Filter, however, Particle Filter popularity is increasing.

Concerning inference methods, there are two main approaches, which are based on Bayesian inference and Dempster-Shafer method. The Bayesian inference based approach allows the information to be fused by applying probability theory rules. By using Bayes theorem, probabilities of *a priori* null hypothesis, a posteriori hypothesis and conditional probability of an observation given a hypothesis can be related. The *a priori* hypothesis probability can be updated by using alternative hypotheses computed basing on observational evidence. The Dempster-Shafer (DS) theory introduces the belief mass, which is assigned to each of possible events. From the mass assignment, a probability interval for an event can be defined. The interval is bounded by belief and plausibility. The DS approach allows supporting the total ignorance about an event, what makes it more reliable than Bayes approach in case of lack of information.

2.5.1 Estimation methods

In the literature concerning environment perception, estimation methods are predominant. There are several methodologies, which use KF as a tool for data fusion and object tracking. KF based fusion methods can be divided into two main groups: measurement fusion model and track-to-track fusion model.

The main idea of the methods of the first group is to fuse the measurements obtained from different sensors and to use fused measurements in the filtering (see Figure 2.7).

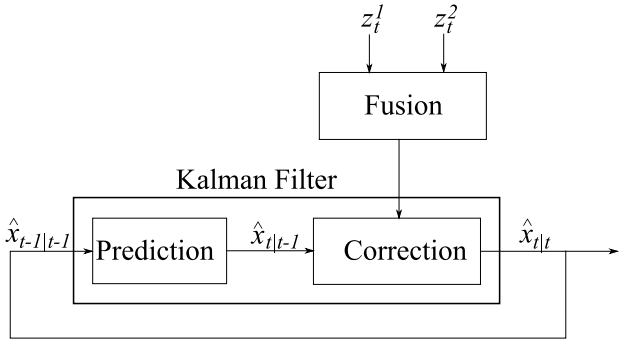

Figure 2.7: Measurement fusion schema.

In this group, two fusion approaches are possible. The first one, called measure merging fusion, is based on the idea of measurements merging in one vector [33]. In this approach, a classical Kalman filtering is used, except from the measurement model, where the measurement matrix H, the measurement vector z and the measurement noise w, with its covariance matrix R, are changed. The measurements z_t^1, z_t^2 are merged into one augmented observation vector [33]:

$$z_t = [(z_t^1)^T (z_t^2)^T] \tag{2.92}$$

The measurement system matrix H becomes:

$$H_t = [(H_t^1)^T (H_t^2)^T] \tag{2.93}$$

and the measurement noise becomes:

$$w_t = [(w_t^1)^T (w_t^2)^T] \tag{2.94}$$

The covariance matrix, under the assumption that sensors signals are independent, is defined as:

$$R_t = \begin{bmatrix} R_t^1 & 0 \\ 0 & R_t^2 \end{bmatrix} \tag{2.95}$$

where R_t^i is the covariance matrix of the ith sensor.

In the second approach, a fused measurement is obtained by the means of a minimum mean square estimate, which combines sensors measurements according to their uncertainty [80]. The measurements fusion equation is expressed as follows:

$$z_t = (z_t^1) + R_t^1(R_t^1 + R_t^2)^{-1}((z_t^2) - (z_t^1)) \tag{2.96}$$

The covariance matrix of the fused measurement can be obtained by :

$$R_t = [(R_t^1)^{-1}(R_t^2)^{-1}]^{-1} \tag{2.97}$$

The fused measurement is then used in a classical Kalman Filter algorithm.

The two measurement fusion approaches are independent from the process noise. They are functionally equivalent if the sensors measurement matrices H_t^1 and H_t^2 are equal [29].

The track-to-track fusion model is introduced in [8, 9]. In this model, the state estimates are obtained for each sensor signal, and then are fused into a new state estimate (see Figure 2.8). The new fused estimate can be calculated by using the following equation:

$$\hat{x}_{t|t} = \hat{x}_{t|t}^1 + [P_{t|t}^1 - P_{t|t}^{12}][P_{t|t}^1 + P_{t|t}^2 - P_{t|t}^{12} - P_{t|t}^{21}]^{-1}(\hat{x}_{t|t}^2 - \hat{x}_{t|t}^1) \tag{2.98}$$

where P_t^i is the covariance matrix of the tracked estimate x_t^i. The covariance matrix of the new estimate is given by the equation:

$$P_{t|t} = P_{t|t}^1 - [P_{t|t}^1 - P_{t|t}^{12}][P_{t|t}^1 + P_{t|t}^2 - P_{t|t}^{12} - P_{t|t}^{21}]^{-1}[P_{t|t}^1 - P_{t|t}^{21}] \tag{2.99}$$

The cross covariance matrix $P_{t|t}^{12} = (P_{t|t}^{12})^T$ is defined by the following recursive equation:

$$P_{t|t}^{12} = (I - K_t^1 H_t^1)F_t P_{t|t-1}^{12} F_t^T (I - K_t^2 H_t^2)^T + Q_t(I - K_t^2 H_t^2)^T \tag{2.100}$$

with the initial condition $P_{0|0}^{12} = 0$.

The advantage of this method is that the object can be tracked simultaneously by local trackers, and then only fusion equations are performed on the central unit. This is a way to distribute the computations. It is shown in [63] that this method gives worse results in terms of uncertainty than the minimum mean square estimate based measurement fusion.

The track-to-track fusion model is very popular. In [16], it is used for obstacle detection in road environment. The authors tested the fusion method

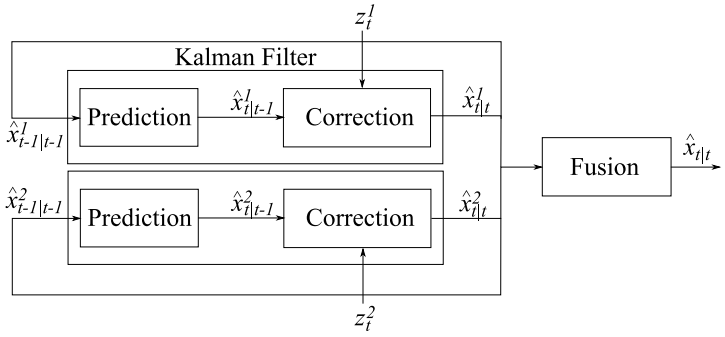

Figure 2.8: Track-to-track fusion schema.

with two sensors configurations. The first one considers an infrared camera and a Radar. The second configuration is built with a Radar and a LIDAR. The obtained results show that the fusion method can overcome the drawbacks of each sensor, with a diminution of the rate of false detection alarm.

In [30], two variants of the track-to-track method are proposed. The first one is called a modified track-to-track method (MTF), and the second variant is called a track fusion model with fused prediction (TFP).

In the first variant, the prediction of each tracked object is not based on the local state estimation $\hat{x}^i_{t|t}$ like in the basic track-to-track method, but it is based on the fused state estimate $\hat{x}_{t|t}$ (see Figure 2.9). Thus, Kalman gains in local trackers are in this case related to the fused prediction state. This is due to the fact that the fused estimate is less uncertain and more accurate than each of the local state estimation. This approach changes the update stage of the track-to-track algorithm.

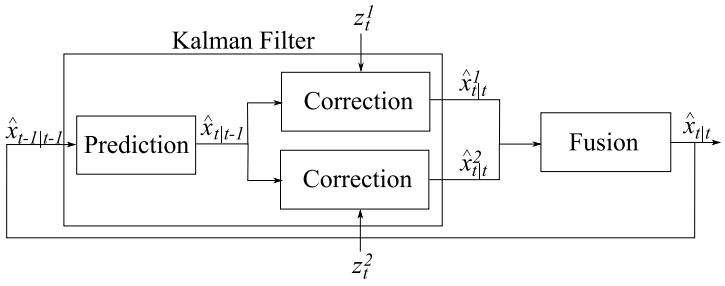

Figure 2.9: Modified track-to-track fusion schema.

The measurement prediction is given then by:

$$z^i_t = H^i_t \hat{x}_{t|t-1} \tag{2.101}$$

The update equation is defined as follows:

$$\hat{x}^i_{t|t} = \hat{x}_{t|t-1} + P^i_{t|t-1}(x,z)P^i_{t|t-1}(z,z)^{-1}[z^i_t - \hat{z}^i_t] \tag{2.102}$$

where $P^{-i}_t(x,z) = E[(x_t - x^i_{t|t-1})(z^i_t - z_{it})]$ and $P^{-i}_t(z,z) = E[(z^i_t - \hat{z}^i_t)(z^i_t - \hat{z}^i_t)]$.

The computation cost of the proposed variant is the same as in the case of the basic track-to-track method. Authors using Monte Carlo based experimentation conclude that in the case of similar sensors, the modified track-to-track method is suboptimal to the original track-to-track and measurement fusion ones. However, for dissimilar sensors, by using the modified track-to-track method, it is possible to obtain better estimation without increasing calculation time.

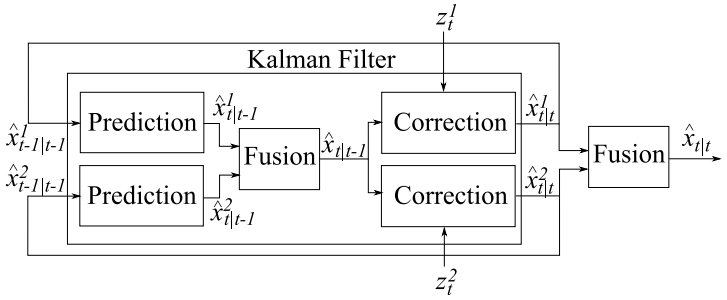

Figure 2.10: Track fusion model with fused prediction schema.

The track fusion model with fused predictions (TFP) introduces the predictions fusion to the standard track-to-track algorithm (see Figure 2.10). The fused prediction $x_{\bar{t}}$ is used to correct the estimates $\hat{x}^1_t$, $\hat{x}^2_t$ with the measurements z^1_t, z^2_t respectively. Finally, the obtained local estimates $\hat{x}^1_t$, $\hat{x}^2_t$ are combined to obtain the fused state estimate $\hat{x}_t$. On the contrary to the modified track-to-track and like in standard track-to-track method, the fused state estimate is stored only on the fusion site. The predictions are obtained by using local state estimates. For details, one can see [30]. Authors claim that this approach has the same performance as MTF algorithm. It may be, however, numerically unstable for similar sensors. In addition, the TFP method is computationally more expensive than the MTF method.

2.5.2 Applications of data fusion methods

In this section, some applications of data fusion techniques in object detection and tracking are presented.

In [36], a hybrid adaptive cruise control system for highways is presented. The radar and vision sensors are used to detect and track the obstacles. Every readout of the radar is pre-processed and the list of potential vehicles is created. For each vehicle hypothesis, an occlusion area is computed. The radars measurements lying in any occlusion area are rejected. Remaining radar measurements, which could not be assigned to an existing vehicle hypothesis or occlusion area, are candidates for new vehicle hypotheses. The candidates are checked by vision. The validated candidates create the tracks. Non validated candidates during several cycles of the algorithm execution are removed. The lane-departure warning functionality is obtained by using information from vision and the ABS system. The vision is used to find lane in the images. The ABS, providing rotation speed of the wheel, allows to compute the vehicle heading. These two information are compared, if needed, to raise the lane-departure alarm.

In [7], a method for inferring scene structure information based on both laser and visual data is proposed. The aim of the proposed system is to provide 3D indoor environment reconstruction in robot motion planning and collision avoidance tasks. The authors use a 2D laser scanner to construct a 3D structure assumptions. This model is then validated by the stereovision data. The validation of the 3D model is achieved by projecting the 3D model points on the two images. The pixels in the two images corresponding to the 3D model point should have the same characteristics such as color, intensity values, intensity gradients. If it is the case, the 3D model is validated, otherwise there exists a strong indication that the 3D model is locally invalid. The normalized crosscorrelation metric [24] is employed to evaluate the correctness of the calculated point correspondences. Low values of the calculated crosscorrelation correspond to regions within the images depicting parts of the environment that are not conform to the 3D model. When the region of inconsistencies between the 3D model and the reality is detected, stereovision is used to provide missing 3D information. The authors [7] proposed also rules for corresponding pixel search based on epipolar constraints. The application of the rules allows to reduce the computation time.

In [28], a three laser scanners configuration is used for objects detection and recognition. The sensors configuration allows to cover the complete surrounding of the vehicle. Two sensors are mounted of the front-left and front-right corner of the vehicle. They allow to cover the area in front of the vehicle as well as the areas along both sides. The third sensor is installed at the back of the vehicle. This sensors spatial configuration im-

proves the object recognition, because gaps in the field of vision of a single laser scanner can be filled with data from other laser scanners. The sensors are synchronized and measure using the common time base. The sensors are calibrated by finding their position on the vehicle using a calibration field. Laser data points coming from different sensors are transformed into the same coordinate system.

In [58], a long range obstacle detection system based on fusion of laser scanner and stereovision is proposed. The main principle is based on the detection and tracking of obstacles by means of a laser scanner while stereovision is used to confirm the laser scanner based detections. The system is exploited for driving assistance purpose. The first step consists of clustering laser scanner data by eliminating the measurements which are outside a warning area. The warning area is defined by a road detection algorithm or by using the heading direction of the vehicle. The vehicle pitch movements may make the laser plane cutting the road surface. This can lead to false detections and losing of the tracked objects. The stereovision based confirmation algorithm consists of four stages: determination of the region of interest in the stereoscopic images, maximization of the detection range by application of a numerical zoom, computation of a local disparity map in the regions of interest (ROI), confirmation of the laser data detection using the computed disparity map. There are three confirmation criteria. In the first criterium, the v-disparity approach [48] is used to classify pixels from ROI into two groups: road-surface pixels and obstacle pixels. Then, the number of the obstacle pixels gives a confidence in the existence of an object over the road surface. This criterion does not make any assumption on the obstacle to be detected. The robustness can be, however, greatly influenced by errors in the disparity map. The second criterium is based on the alignment finding of the observed surface. This is done by using Hough transform. The vertical alignment corresponds to the obstacle and the vertical one corresponds to the road surface. This criterium is less fragile to the disparity map imperfections. In the third criterium the altitude of the laser points are compared with the local road profile, which is estimated by the v-disparity method. The obstacle is confirmed if the altitude of the laser points is greater that a certain threshold.

In the literature no detailed fusion methodology for laser scanners fusion exists. In [28] the multi laser scanner configuration is used. The authors, however, do not present how the raw data points are used in the process object detection and tracking. Thus a method for multiple laser scanners fusion which is adapted to callipers based OBB extraction algorithm is

presented.

2.6 Stereovision

One of the aims of computer vision is to create the three-dimensional structure of a scene from one or more images. This function can be fulfilled by using stereovision. It consists of reconstructing a three-dimensional scene from several images taken under different angles of view. Like the human vision, the perception of depth is achieved by operating the discrepancy (later called disparity) between the images of a same pair. Most techniques rely on the binocular stereovision using only two images. However, trinocular stereovision using three images to remove ambiguities due to the occlusion problem is sometimes used. In this work, we focus on binocular stereovision, and use left and right images to produce a disparity map.

To accomplish 3D reconstruction, four major steps must be considered:

- Calibration of the stereoscopic sensor.
- Extraction of relevant primitives of the image.
- Matching primitives extracted from stereoscopic images.
- Triangulation and 3D reconstruction of the scene.

2.6.1 Principle of stereovision

With a binocular stereoscopic sensor, the two cameras must observe the same scene. It means that their optical axes must converge towards the same scene. It is important to set the cameras to obtain a common angle of view. This constraint is usually non-existent when a pre-manufactured stereoscopic sensor is used. The geometric model of a binocular stereoscopic sensor, presented by Chambon [19], is illustrated in Figure 2.11. By considering the pinhole camera model, a point P of the scene that can be seen in both images (left and right of the stereoscopic sensor) is projected at two points P_g (point in the left image) and P_d (point in the right image). These two points are called homologous points, since they correspond to the same point in the observed scene. Thus, one can determine the equations of the straight line P_gO_g (respectively P_dO_d), which goes through the optical center of the left camera O_g (respectively O_d) and the point P_g (respectively P_d). These equations must be expressed in a common reference

system (usually the absolute reference system associated to the scene). It is therefore necessary to determine transformations between the 3D space of the scene and the 2D space of the left and right images. These transformations are generally obtained in the calibration step of the stereoscopic system. Finally, by geometric triangulation, the intersection of these two straight lines P_gO_g and P_dO_d gives the position of the point P of the real scene.

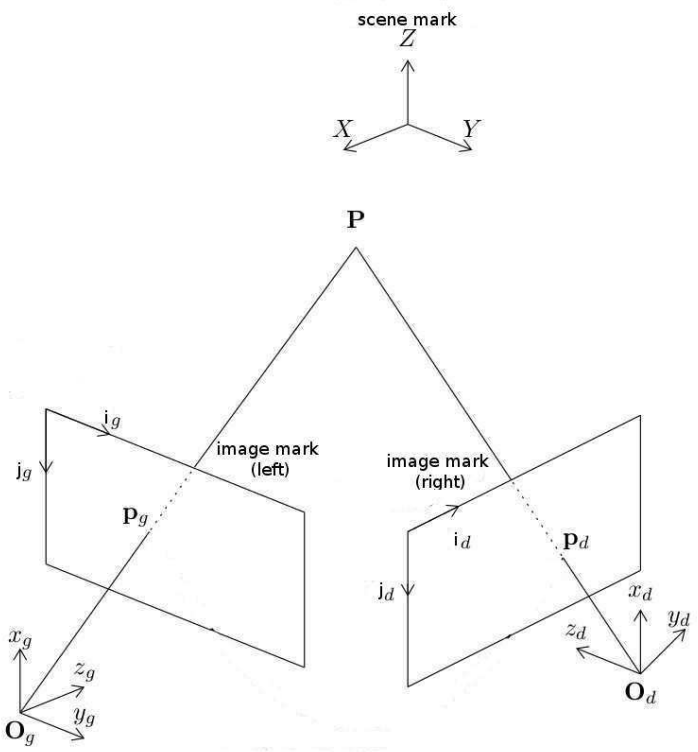

Figure 2.11: Geometric model of a binocular stereoscopic sensor.

2.6.2 Calibration of a stereoscopic sensor

Calibration of a stereoscopic sensor consists of determining two geometric transformations (one of each camera) between the 3D space of the scene and the 2D space of the left and right images. These transformations are determined using extrinsic parameters (position and orientation of each camera relative to the 3D reference system) and intrinsic parameters of each camera (focal length, size and resolution of the sensors). For each camera, the associated transformation is composed with two parts. The first one allows

going from the absolute reference system to the camera relative reference system. The second part expresses the projection process of a 3D point in the image of the camera. The combination of these two parts, given generally by two matrices, allows to calibrate the system.

2.6.3 Extraction of primitives

Extraction of primitives consists of extracting visual cues characterizing objects in each image. Obviously, primitives must be enough abundant and have attributes that allow them to be discriminated in order to remove ambiguities during the matching step. Several types of primitives can be considered: interest points (Moravec points[55], high curvature points, SIFT points[50], etc.), edges, snakes[44] and regions.

2.6.4 Stereo matching

The key problem in stereovision is the matching task, which consists in comparing each feature extracted from one image with a number, generally large, of features extracted from the other image in order to find the corresponding one, if any. This process, which is difficult to perform, requires a lot of computation, as well as a large amount of memory [12]. Once the matching is established and the stereo vision system parameters are known, the depth computation is reduced to a simple triangulation technique [39, 23].

Many approaches have been proposed to solve the stereo matching problem. According to the considered application, the existing techniques are roughly grouped into two categories: area-based and feature-based [32]. Area-based methods use correlation between brightness patterns in the local neighbourhood of a pixel in one image with brightness patterns in the local neighbourhood of the other image [66]. These methods lead to a dense depth map. Feature-based methods use zero-crossing points, edges, line segments, etc. and compare their attributes to find the corresponding features [53, 13, 20, 39, 66]. These methods lead to a sparse depth map. To resolve matching ambiguities, feature-based and area-based methods use some constraints like epipolar, uniqueness, smoothness and ordering [71, 52, 77, 64].

Chapter 3

Thesis statement

A system for dynamic objects tracking can be divided into following main stages: perception, pre-processing, detection and clustering, association, filtering (see Figure 3.1). The perception part is responsible for providing observation data of the environment. The pre-processing prepares the data to be treated in the following stages. The third stage consists of object detection and data clustering. The next stage, data association, is responsible for correct correlation of measurements, extracted from observation data, with object tracks. Finally, the filtering part responsible for objects state estimation and prediction. The filtering part provides information, needed for fulfilling higher level task such as driving assistance, navigation, platooning.

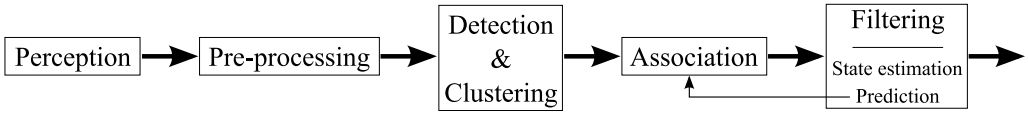

Figure 3.1: Schematic representation of a dynamic objects tracking system.

Basing on the analysis of the existing algorithms and systems, the following thesis statement is proposed:
Increasing the reliability and robustness of tracking system by introducing an OBB based representation relaxing linear surface assumption, an Inter-Rays uncertainty, a Fixed Size assumption, a two laser scanner fusion algorithm, a data association algorithm based on Nearest-Neighbour princi-

ple and the Fixed Size assumption, a stereo-vision and laser scanner fusion for data clustering.

The statement can be decomposed into five arguments :

- The usage of the OBB representation model and OBB extraction algorithm, based on rotating calipers and on-line convex-hull creation technique is adequate to tracking of dynamic objects. The proposed representation model and algorithms relax linear surface assumption, increase the measurement accuracy and do not need all data points to be considered.

- The Inter-Rays (IR) uncertainty and Fixes Size (FS) assumption increase the quality of object size and position estimation.

- The two laser scanner fusion algorithm increases tracking accuracy in terms of object size, angle orientation, velocity estimation.

- Based on the Nearest-Neighbour algorithm and the Fixed Size assumption, the data association algorithm for the OBB representation model increases the robustness of the tracking by a reliable data association for coalescing objects.

- The laser scanner and stereo-vision fusion algorithm allows to cluster correctly laser data points in presence of ambiguous point configurations.

As stated in the introduction, the thesis is a part of the project "Intelligent Vehicles and their integration in the city of the future". The objective of this project is to ensure a vehicle autonomous navigation in an urban environment. One of the most important and fundamental part of the autonomous vehicle navigation is perception of the environment. The environment can be treated as a decomposition of different entities such as: ground, obstacles (static objects), ground markings, road signs, traffic lights, and dynamic objects. Construction of a system, which will be able to track dynamic objects is the scope of this thesis. The dynamic objects perception system is composed of different algorithms and models, which allow to solve the following problematics: object detection, object representation, data association and tracking.

In the dissertation, an Oriented Bounding Box (OBB) model is proposed. The main idea of the proposed approach is to provide a representation model that does not depend on the assumption that objects are constructed

by linear surfaces, what is the case of the majority of based line fitting algorithms. The fundamental part of the OBB based approach is creation of a Convex-Hull (CH) for laser data points assigned to the object. The CH can be obtained by using an on-line algorithm adapted to laser scanner data or by using a line tracking algorithm. The usage of the first algorithm allows to relax the aforementioned assumption , without a decrease in tracking accuracy. In addition, it is less computational complex compared to the line tracking algorithm. The proposed OBB extraction method increases measurement accuracy by guaranteeing extraction of the best aligned OBB. The first part, the most computationally complex, of the proposed solution does not need all data points as in most of line fitting algorithms.

The Inter-Rays (IR) uncertainty is integrated into the proposed OBB extraction algorithm. It introduces additional position and size uncertainties taking into account the laser sensor characteristics, i.e. the fact that extreme raw data points constructing an OBB do not represent the object's real extremities. Finally, the Fixed Size assumption based on the fact that in general tracked objects do not change their size. The FS assumption allows to store the best, in terms of uncertainty, size estimation obtained during the tracking, and thus, increases the precision of the position and size estimation.

The two laser scanner fusion algorithm takes advantage of angular resolution complementarity characteristics of two laser scanners configuration. The data points of the two laser sensors, merged into one cluster, allow to obtain more accurate OBB measurement. The two sensors configuration allows also to increase the velocity estimation reliability by eliminating oscillations appearing in the velocity value when one sensor configuration is used.

The data association stage allows to correct object tracking. In the thesis, a data association algorithm, exploiting the geometrical representation of the object, is proposed. It is based on the NN principle and FS assumption. The idea of the algorithm is that only points, which do not violate the FS assumption of the track, can be assigned to it. If a point violates the FS assumption of all probable tracks, the decision to which track it should be assigned is based on the NN principle. The proposed method gives reliable coalescing object separation, even for objects touching each other.

Finally, the sensor fusion algorithm for laser points clustering algorithm is presented. In the algorithm, stereovision based 3D information is used to answer the question if two laser data points belong to the same object. To obtain the answer, the discontinuity in the disparity map, extracted from

two images, is seek between the two image points obtained by projecting the laser points onto the stereo images. The proposed solution gives correct results in presence of ambiguous laser data points configurations, for which only laser data based algorithms fail.

The statements of the thesis will be proved by constructing the perception system, incorporating the proposed solutions, and by evaluating and testing the algorithms characteristics on simulated and real vehicle platforms. The simulator, which is a part of the thesis work, is developed on the purpose of evaluating and testing the proposed algorithms. The real vehicle platform is a part of the "Intelligent Vehicle" project, led by the Systems and Transportation laboratory of the University of Technology of Belfort-Montbeliard and supported by the Regional Council of Franche Comt and the Contrat Projet Etat Région

Chapter 4

Testbed

4.1 Simulator

As a part of the "intelligent vehicles and their integration in the city of the future" project, a software platform is developed to simulate the sensors and the multiple objects tracking process. The simulator permits flexible changing of all sensors parameters and mounting position. In the simulator, laser range finder (LRF), LIDAR, stereovision and odometry sensors are implemented. This allows to test the developed algorithms with different sensor configurations. The simulator generates data from each sensors configuration by playing virtual scenarios, which can be visualized in real-time (see Figure 4.1).

The simulator development is based on an Agile software development methodology. Agile methods break tasks into small increments with minimal planning, and do not directly involve long-term planning. This methodology is chosen since, in our opinion, it is very well suited for producing prototypes.

The simulator allows to:

- Visualize 3D worlds.
- Simulate dynamic objects.
- Simulate sensors such as: 2D,3D laser scanner sensors, including systematic and statistical errors, mono and stereovision sensors and odometry.
- Construct 3D worlds by using 3D models with textures.

- Construct itinerary paths to be followed by the vehicles, including the ego-vehicle.
- Save and load a created scenario and algorithms parameters.

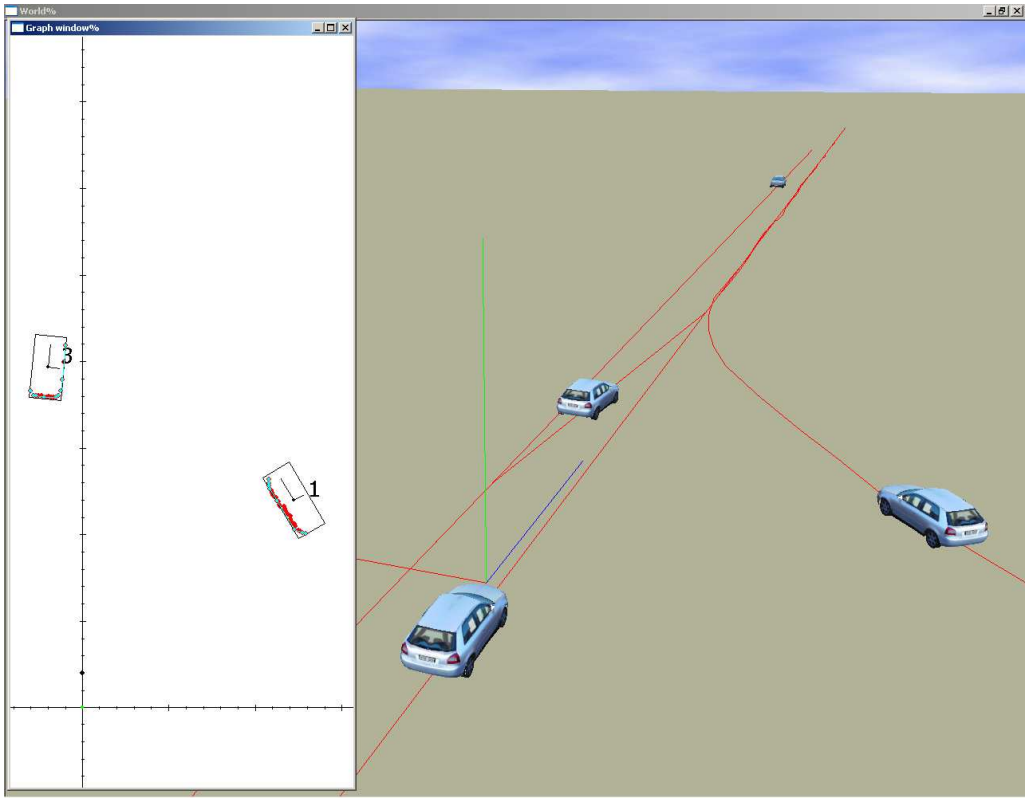

Figure 4.1: Simulator - screenshot.

The tools used for the simulator development are chosen by taking into account the following aspects: compatibility with C++, free usage, access to the source, portability, performance, documentation and community support and stability. All the used tools are under licenses that grant free usage, and access to the source. The feature of the open source software is that, very often, the documentation is not up to date or misses important information. In such situation, the community support is very important. Due to this facts, the development of the simulator can be viewed as a research activity because it as demanded to find new technical solutions. Nevertheless, thanks to the free to use software, it is possible to create necessary tools with limited funding.

The software (IDE) and used libraries are:

- Object-Oriented Graphics Rendering Engine (OGRE), written in C++, is a scene-oriented 3D engine. The library simplifies production of applications utilising hardware-accelerated 3D graphics. The library is based on the two most popular 3D graphics system libraries: Direct3D and OpenGL. The details of these system libraries are hidden thanks to abstract classes included in OGRE. A very important aspect is that OGRE is released under an open source license: GNU Lesser Public License (LGPL).

- Open Dynamic Engine (ODE) (open source) is a high performance library for simulating rigid body dynamics. It is fully featured, stable, mature and platform independent, with an easy to use C/C++ API. It has advanced joint types and integrated collision detection with friction. ODE is useful for simulating vehicles, objects in virtual reality environments and virtual creatures. It is currently used in many computer games, 3D authoring tools and simulation tools.

- OgreOde is an ODE wrapper for OGRE. The wrapper simplifies the process of implementation, and includes some prefabricated objects like vehicles or ragdolls.

- wxWidgets is an API for writing GUI applications on multiple platforms that still utilize the native platform's controls and utilities. It allows to link with the appropriate library for different platforms like Windows, Unix, Mac OS.

- GNU Scientific Library (GSL) is a numerical library for C and C++ programmers. The library provides a wide range of mathematical routines such as random number generators, special functions and least-squares fitting.

- Code::Blocks (open-source) is a cross-platform C++ integrated development environment (IDE).

- Blender (open source) is a cross-platform suite of tools for 3D creation.

- Gimp - GNU Image Manipulation Program (free software) is a raster graphics editor.

- Subversion (SVN) is an open-source revision control system.

In the test we use different vehicles models, which are downloaded from internet, and prepared to be used in the Blender. For the purpose of FS

evaluation, a special vehicle model is prepared. It is prepared in such manner that the profile obtained by cutting the model by laser the plane has the maximal vehicle observable size. To obtain it, the mirrors are removed and the vertical profile is changed (see Figures 4.2 and 4.3). This is done to assure that maximal vehicle size is detectable by a laser scanner.

Figure 4.2: Deformed 3d model of the vehicle used in FS algorithm evaluation.

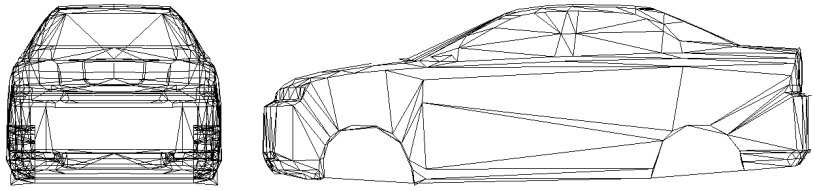

Figure 4.3: Deformed 3d model of the vehicle used in FS algorithm evaluation (wireframe).

4.2 Real vehicle platform

The first research platform, acquired by the laboratory, is an automotive robot car, called robuCAB and manufactured by RoboSoft [1]. It is equipped with a SICK LMS laser scanner, sonar, and an embedded computer. The main disadvantage of this platform is that it is not authorized to be present on public roads. Thus, decision is made to create a new research platform, which will not have the drawbacks of its ancestor. The new platform is assembled by the research team of the SeT laboratory (see

Figure 4.4). The base of the platform is an electric car designed by Global Electric Motorcars (GEM), a subsidiary company of Daimler Chrystler. The commercial name of this car is GEM e2. This car has been designed and developed exclusively for an urban usage. The GEM e2 is equipped with all necessary safety devices and is authorized on public roads. The vehicle runs with the maximal speed of 45 km/h. Its range on fully recharged battery is 50 km.

The control system of the platform is installed by specialized team of the SeT laboratory. Thanks to this installation, the vehicle can be driven by a person or controlled by a computer. The driving commands send by the computer are converted to the actuators by a MicroAutoBox, which is a real-time system for performing fast function prototyping. The MicroAutoBox is a component widely used for designing and testing different kinds of prototypes. It is, for example, used in automotive and avionic research. This module is very expensive, thus, it is planned to replace it by a micro control cards, which are much cheaper. But, for doing this, the control software must be optimized.

Figure 4.4: Real vehicle platform.

The platform must be able to perceive its surrounding environment and to localize itself in it. Thus, the installed sensors can be divided into two groups: the perception sensors and the global positioning and attitude sensors. The first part consists of three sensors: a vision sensor and two laser scanners. The first sensor is a Bumblebee2 stereoscopic color cameras which allow to obtain depth information of a scene. The stereoscopic sensor con-

sists of two 1/3 inch CCD sensors and 3.8 mm focal length lens, what give a horizontal field of view of 66°. The maximal resolution is 1024 x 768 pixels with a frame rate of 20 images per second. The first laser scanner is a SICK LMS 221, which provides 2D data. It a maximum of 180° for the horizontal field of view. The angular resolution is set to 0.25°, 0.5° and 1° depending on the used horizontal field of view. The maximal detection range of 80 meters. It can operate in two modes: millimeter mode and centimeter mode. The first one provides more accurate measurements, but its range is limited to 8 meters. The other mode allows to detect objects with a range of 80 meters. The second laser scanner is an Ibeo LUX lidar. It allows to obtain 3D data providing 100° horizontal and 3.2° vertical field of views. The maximum detection range is 200 meters. The angular resolutions can be set between 0.125° and 1°, depending on the size of the field of view. The detection range of the laser scanners depends on the surface reflection characteristics and lightning conditions.

The global positioning of the experimental vehicle is obtained by a GPS-RTK (ProFlex 500 Magellan). The Real Time Kinematic technology (RTK) gives a global position with a much greater accuracy compared to standard GPS. The standard deviation sigma of the horizontal (respectively vertical) measurement uncertainty is 1 centimeter (respectively 2cm). The attitude of the vehicle is measured by a Mti Xsens sensor. The Mti Xsens contains a 3D gyroscope, an accelerometer and a magnetometer. A real-time proprietary sensor fusion algorithm, running on an internal low-power digital signal processor, provides drift-free 3D orientation data.

To calculate the real vehicle positions, three GPS-RTK are used. The first one plays a role of the reference base. It is fixed and, hence, it can provide RTK corrections. The second one is installed on the experimental vehicle. The third one is installed on the tracked vehicle. The data obtained from the two GPS sensors installed on the vehicles allow to calculate the position, velocity and orientation for the tracked vehicle. The orientation angle can also be approximated, but it does not represent the real vehicle heading since it is calculated by using the current and last vehicle position points. In the calculations, the GPS-RTK position uncertainty is taken into account.

Chapter 5

Object representation and tracking

5.1 Oriented bounding box

Point and ellipse based object representation are not suitable for the problem being treated, ie. tracking objects in urban environment. The two mentioned models may be accurate enough in simplified environments such as highways, where:

- distances between objects are much greater than their size
- there are important movement constraints (limited direction, limited maneuvering), what implies more predictable objects behaviour
- there is a limited types of objects which can be met, what results in almost uniform object's shape
- a surface is discretised by lanes

The characteristics of such environments imply that the ego vehicle can keep big distances to the other objects, and path planning can be reduced to simple tasks (for example stay in the lane / change the lane). In this kind of environments it is very important to detect far objects, what gives long time enough for navigation algorithm to react to a new situation. Navigation algorithm, however, can be reduced to simple tasks such as accelerate, decelerate, stay in the lane, change the lane. On may say that navigation of vehicles in those environments has reduced degrees of freedom.

In diverse environments, however, such as cities, one needs a geometrical representation or at least an approximation of the obstacle's geometry. This is due to:

- short distances (smaller than object's size).
- one cannot be sure that space is discretised.
- little space available for navigation: the ego vehicle needs to execute path planing frequently. There is also a need of usage of each free space.
- little movement constraints: it is difficult to predict objects movements, which can be predicted in far shorter run than in the case of highway environments.
- there are vast kinds of objects moving around.
- necessity of more accurate data association, what may be really difficult without objects shape and size approximation.

Navigation in diverse environments is a very demanding task. There are little constraints on objects manoeuvres. There are driving rules. However, one can expect everything from the other objects moving around (different speeds, different directions, short distances between the objects).

The rectangle based representation is suitable for representing objects in demanding environments such as urban areas. The advantages of the rectangle based object's representation:

- shape approximation, and thus, space occupation description
- adequate precision
- easiness of position's and size's uncertainties representation and computation
- simplicity and intuitive usage
- data compression

In the literature the proposed algorithms assume, however, that objects have a rectangular shape (or eventually a shape of rectangle with rounded corners). In Oriented Bounding Box (OBB) based approaches this assumption is relaxed, and thus, the OBB model introduces a uniform representation for objects with different shapes.

It is important to notice that a vast majority of dynamic objects in urban environments are of the types: personal car, bus, human, bicycle/motorbike, one part trucks. Each of them is convex and can be represented by an OBB with a sufficient accuracy. Of course, there are also more complicated concave objects. But, from the navigation point of view, these objects can be divided into two groups. The first one contains compact objects whose concavity cannot be treated as a free space, and so, they can be represented by an OBB without a lost of accuracy. The other group comprises complex objects whose concavity is a navigable area or must be taken into account to produce better predictions, (for example an articulated buse or truck) (see Figure 5.1). The objects from the second group cannot be represented by an OBB, but multiple OBB can be used instead. Static obstacles can be represented using the same principle. Static obstacles can be dividing into two groups. The first one consists of objects which can be represented by one OBB. In in the second group, there are objects which have to be represented by multiple OBB. If a space obtained by a subtraction between an OBB and the actual object shape is navigable the object belongs to the first group; otherwise, it belongs to the second one.

An OBB based measurement is described by a state vector z including centre coordinations cx, cy, orientation angle α, size dx, dy and an uncertainty vector σ_z:

$$z = [cx, cy, \alpha, dx, dy]^T \tag{5.1}$$

$$\sigma_z = [\sigma_{cx}, \sigma_{cy}, \sigma_\alpha, \sigma_{dx}, \sigma_{dy}]^T \tag{5.2}$$

An OBB based track is described by a state vector x and an uncertainty vector σ_x, which include the linear velocity $\dot{cx}$, $\dot{cy}$ and the angular velocity $\dot{\alpha}$, in addition to the measurement vectors z and σ_z:

$$x = [cx, \dot{cx}, cy, \dot{cy}, \alpha, \dot{\alpha}, dx, dy]^T \tag{5.3}$$

$$\sigma_x = [\sigma_{cx}, \sigma_{\dot{cx}}, \sigma_{cy}, \sigma_{\dot{cy}}, \sigma_\alpha, \sigma_{\dot{\alpha}}, \sigma_{dx}, \sigma_{dy}]^T \tag{5.4}$$

The OBB construction is executed after or during data association stage. It depends on the data association strategy. Nevertheless, the OBB is extracted from laser scanner raw data points grouped in a cluster.

The presented method of the OBB construction consists of two main steps. In the first step, creation of an object's convex contour takes place. In the second step, a method of Rotating Calipers [73] is used to construct an OBB, which is the best aligned to the object's contour.

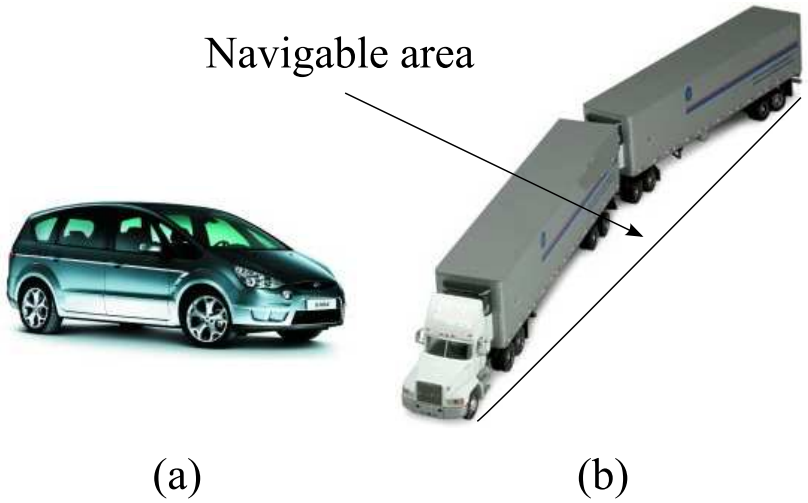

Figure 5.1: (a) Compact object (b) complex object.

5.1.1 Convex contour construction

The convex contour is in fact an open convex-hull. This is due to the fact that only a part of an object is visible. Below, there are two different approaches to open convex-hull construction. Both algorithms operate on raw data points. The first one is direct on-line open convex-hull creation algorithm. In the second algorithm, one convex-hull is obtained by finding intersections of extracted points line segments. The raw data points can be produced by a laser range finder (LRF), lidar, or stereovision (SV). The algorithms, however, are adapted to be used with 2D or 3D lidars since they achieve calculations on polar coordinates and assume that input points are sorted with increasing order of the horizontal angle coordinate. This feature of the algorithms does not exclude stereovism as a data source, though the data must be sorted. The problem of polar coordinates may be resolved in two ways: either data are converted from Cartesian to polar coordinates system or the algorithms should be reimplemented so they operate on Cartesian coordinates.

On-line convex-hull creation

The algorithm is the based on sequencing characteristic of the raw scan points. In the beginning, two first points are added directly to the convex-hull. For each next point added to the convex-hull, the convexity test is executed. To explain the convexity test, let us use two examples (see Figure 5.2). The examples show a convex-hull, which is described by the points

D, C, B and A. It is assumed that the point O represents the origin of the sensor's coordinate system. When a new point N is considered, two cases can be distinguished. In the first one (see Figure 5.2 (a)), the point N can be added to the existing convex-hull by connecting it with the point A, without violating the convexity constraint. The second case takes place (see Figure 5.2 (b)), when by adding the point N, the convexity constraint is broken. In this case, connecting the points N and A produces a concavity represented by NAB. To recognise the two mentioned cases, the proposed algorithm computes and compares lengths of the two line segments OP and OA, where P is the intersection of the lines NB and OA. If the length OP is greater than OA, the point N is added to the convex-hull and the next iteration of the algorithm takes place. When the length OP is less or equal to OA, the point A is removed and the convexity test is repeated for the remaining points constructing the convex-hull (in the example: B, C, D). The repetition of the test is stopped when the convexity condition is not violated for the point N and the two last points in the convex-hull (like in example (a)).

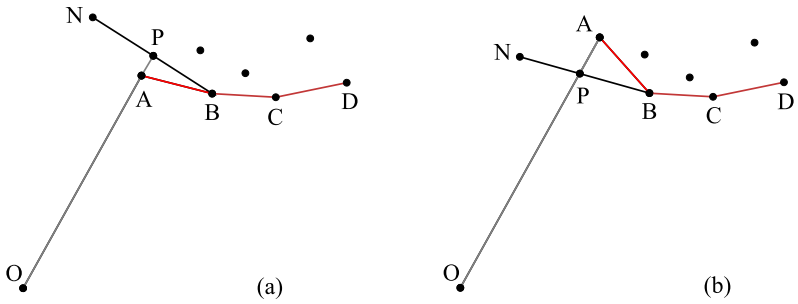

Figure 5.2: Convex-hull construction.

Line segment extraction based convex-hull creation

The second algorithm for creating a convex-hull is based on line segments, extracted from raw data points.

Basing on the comparison of the different line extraction algorithms available in the literature [56, 68], a method called "Line tracking" (LT) is chosen. The advantages of this algorithm are the simplicity as well as the performance in terms of reliability. The only algorithm that outperforms LT is the "Split-and-Merge" (S-M) algorithm [57]. However, LT has one important advantage over S-M. LT algorithm does not need all raw data points to be executed [68]. LT can run in incremental manner and that is

why it may be parallelized with data association stage. The LT algorithm computes a line segment for successively incoming points by the means of linear regression. The algorithm starts with a segment estimated from the first two points. For each next incoming point, a distance condition is checked (see Figure 5.3). If the distance of the new point ($P6$ in the Figure 5.3) to the estimated line ($Seg1$ int the Figure 5.3) is below a certain threshold, a new line including the new point is computed. If the distance is over the threshold, the line is saved as a line segment and the algorithm restarts by estimating a new line. Construction of the line starts from the last point of the saved line segment and the new point, which has not meet the distance condition. The distance threshold is based on the sensor accuracy and uncertainty of the line estimation. The distance condition is expressed as follows:

$$T \geq D_N = \frac{d}{\sigma} \tag{5.5}$$

Where D_N represents the normalised distance between the new point and the line, σ is the sum of the line's radii σ_r and angle uncertainties σ_α, and the point range uncertainty σ_p projected on the line:

$$\sigma = \sigma_r + \sigma_\alpha + \sigma_p \tag{5.6}$$

$$\sigma_p = \sigma_\rho \cos\beta \tag{5.7}$$

where β is the difference between the line's angle and the point's angle in the polar coordinate system, σ_ρ is the sensor's range uncertainty. The threshold can be found as a square root of the value from χ^2 distribution with 1 degree of freedom and with a probability that a correct measurement (a new point) belongs to the line.

In the proposed method, weighted total-linear-regression technique, incorporating the laser range finder sensor model, is used [5]. Considering the laser range finder, it is assumed that the sensor has only range uncertainty, while angular uncertainty is neglected. The equations for finding the line orientation angle α, radii r and their uncertainties σ_α and σ_r are expressed as follows:

$$\tan 2\alpha = \frac{-2\sum(\bar{y}_w - y_i)(\bar{x}_w - x_i)}{\sum[(\bar{y}_w - y_i)^2 - (\bar{x}_w - x_i)^2]} \tag{5.8}$$

$$r = \bar{x}_w \cos\alpha + \bar{y}_w \sin\alpha \tag{5.9}$$

where $\bar{x}_w = \frac{1}{N}\sum \rho_i \cos\theta_i$ and $\bar{y}_w = \frac{1}{N}\sum \rho_i \sin\theta_i$. (ρ_i, θ_i) is the ith point's coordinates in the polar system, (x_i, y_i) is the ith point's coordinates in the

Cartesian system.

$$\sigma_\alpha^2 = \frac{1}{(D^2+N^2)^2} \sum w_i^2 (A-B)^2 \sigma_{\rho_i}^2 \tag{5.10}$$

$$\sigma_r^2 = \sum [\frac{w_i}{\sum w_j} cos(\theta_i - \alpha) + \frac{\partial \alpha}{\partial \rho_i} (\bar{y}_w \cos\alpha - \bar{x}_w \sin\alpha)]^2 \sigma_{\rho_i}^2 \tag{5.11}$$

where $\frac{\partial \alpha}{\partial \rho_i} = \frac{A-B}{D^2+N^2}$, $A = N(\bar{x}_w \cos\theta_i - \bar{y}_w \sin\theta_i - \rho_i \cos 2\theta_i)$
and $B = D(\bar{x}_w \sin\theta_i + \bar{y}_w \cos\theta_i - \rho_i \sin 2\theta_i)$, D and N are the denominator and the numerator of the right hand side of the equation (5.8) respectively. w_i is the weight associated to the ith point. $\sigma_{\rho_i}^2$ is the sensor's range variance associated with the ith point. In the proposed method, $\sigma_{\rho_i}^2$ has the same value for each point and is equal to σ_ρ^2.

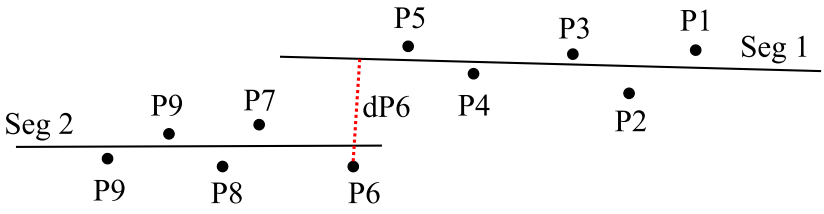

Figure 5.3: Line tracking.

The execution of the LT based method provides a set of lines describing a contour. The next stage of the OBB construction algorithm needs a convex-hull, which is represented by a list of points. To convert the line based contour description to a convex-hull, the intersection points of the extracted lines and the extreme data points of the first and last extracted line segments are considered. There are cases, especially for a big sensor's range uncertainty, where the intersection of consecutive line segments is outside the Axis Aligned Bounding Box (AABB) of the raw data points. Taking into account those points in the process of convex-hull construction gives erroneous results. To avoid those situations, the intersection of the problematic lines with the AABB sides are used to create the convex-hull, instead of the lines intersection situated outside the AABB.

5.1.2 Finding the best aligned OBB for the convex contour

To find the best aligned OBB, rotating calipers (RC) algorithm [73] is used. The original RC algorithm operates on closed convex-hulls and uses four calipers. In our case, however, only open convex-hulls are available. Results of the direct execution of the RC algorithm using an open convex-hull are not reliable (see Figure 5.4). This is due to the fact that the RC algorithm takes also into account the edge connecting the first and the last points in the open convex-hull. This edge cannot be considered since it represents the invisible part of the object. Hence, the orientation angle, which does not correspond to the visible part, is obtained.

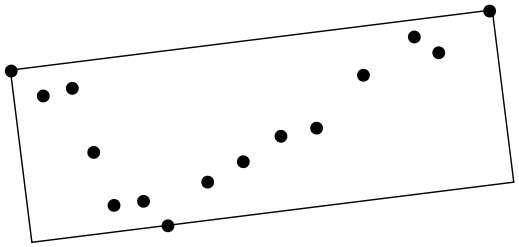

Figure 5.4: Case when the direct execution of the RC algorithm fails.

To avoid this problem, it is assumed that the contour of visible and invisible parts of the object are symmetrical with respect to the point, which is the middle of the segment defined by the two extremities of the constructed contour (see Figure 5.5). Thanks to the symmetry assumption, it is possible to use only two calipers.

In the algorithm, it is assumed that the contour of visible and invisible parts of the object are symmetrical with respect to the point, which is the middle of the segment defined by the two extremities of the constructed contour (see Figure 5.5). The rotating calipers (RC) algorithm begins by bounding the contour through its extreme points using four lines determining a rectangle (in our case 2 perpendicular lines). In each step of the algorithm, at least one of these lines coincides with one of the edges of the contour. The lines are simultaneously rotated in one direction, about their supporting points ($P2$ and $P4$ in Figure 5.5) during each iteration of the algorithm. The rotation angle has a value, which permits for one of the lines to coincide with the next edge of the contour (in the Figure 5.5 the lines are rotated by the angle α). For each lines' position, an area of bounding rectangle, created by four lines (two lines, in our case, $C1$, $C2$ and their

symmetrical lines), are computed. This is performed by computing the area of the rectangle defined by the line segments $MM1$ and $MM2$, where M is the middle of the line segment defined by the extreme open convex-hull points $P1$ and $P5$. $M1$ and $M2$ are respectively the intersections of the lines $C1$ and $C2$ with theirs perpendicular lines passing by the point M. The process is repeated until reaching the expected rotation angle. The smallest area over all iterations indicates the orientation angle θ of the minimum-area OBB, which is at the same time the best aligned one.

During the execution of the RC algorithm for each rectangle (two lines rotation), the two end points of the convex-hull's edge coinciding with the line are saved. These points are used to compute the orientation angle uncertainty.

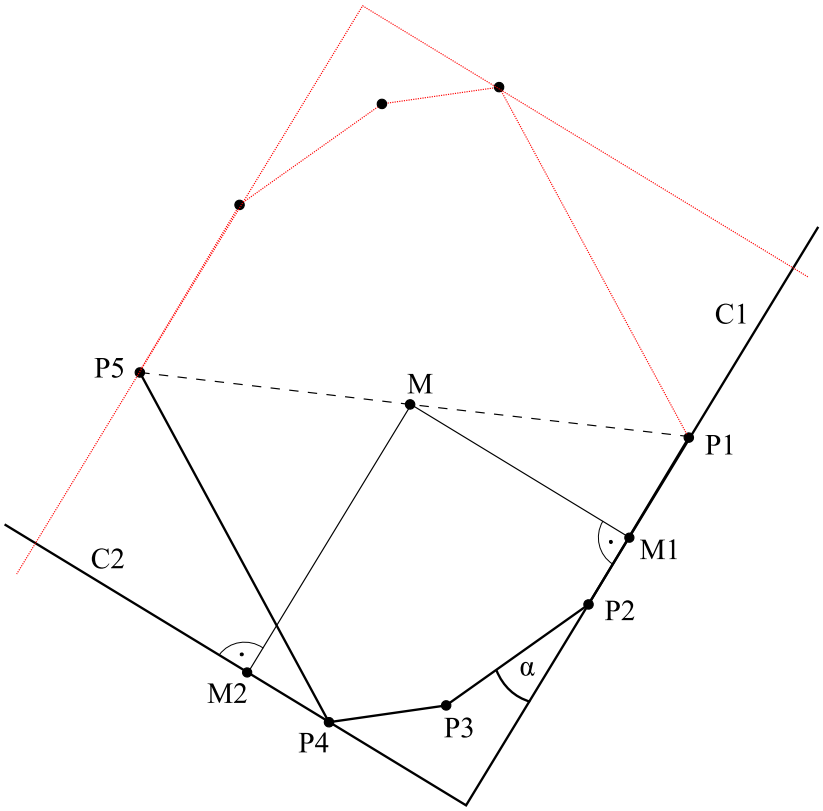

Figure 5.5: Rotating calipers.

To find the size of the OBB, points constructing the convex-hull are transformed to the OBB's local coordinate system. The origin of the OBB's local coordinate system is located at the point M, which is the convex-hull symmetry point. The angle α defines the rotation angle of the OBB's local coordinate system.

5.1.3 OBB uncertainties

After extracting the measurement OBB, the uncertainty vector must be found. The vector consists of covariances of the centre position, size and orientation angle.

The centre and size uncertainties, given respectively by equation (5.12), are found by applying an error propagation law [3].

$$\begin{aligned} \sigma_{cX}^2 &= \tfrac{1}{4}\sigma_{dX}^2 \\ \sigma_{cY}^2 &= \tfrac{1}{4}\sigma_{dY}^2 \\ \sigma_{dX}^2 &= \sigma_{minX}^2 + \sigma_{maxX}^2 \\ \sigma_{dY}^2 &= \sigma_{minY}^2 + \sigma_{maxY}^2 \end{aligned} \tag{5.12}$$

where σ_{minX}^2, σ_{maxX}^2, σ_{minX}^2 and σ_{maxX}^2 are defined as follows:

$$\begin{aligned} \sigma_{minX}^2 &= |\sigma_\rho^2 \sin\beta_{minX}| \\ \sigma_{minY}^2 &= |\sigma_\rho^2 \cos\beta_{minY}| \\ \sigma_{maxX}^2 &= |\sigma_\rho^2 \sin\beta_{maxX}| \\ \sigma_{maxY}^2 &= |\sigma_\rho^2 \cos\beta_{maxY}| \end{aligned} \tag{5.13}$$

with β_{minX}, β_{minY}, β_{maxX} and β_{maxY} are respectively the angles of the points with the minimum x coordinate, minimum y coordinate, maximum x coordinate, the maximum y coordinate in the constructed OBB (see Figure 5.6). The angles β are defined between the Y local axis and the laser ray passing by the extreme OBB points for which the angle is defined. Since the center position uncertainty is computed from the size uncertainties, it is related to the local OBB coordinate system. Thus, σ_{cX}^2 and σ_{cY}^2 have to be rotated:

$$M_r = M_l * R \tag{5.14}$$

where M_l and M_r are respectively the center position covariances matrices respectively in the local OBB and ego-vehicle coordinate systems.

The matrix M_l is defined as follows:

$$\begin{bmatrix} \sigma_{cX}^2 & 0 \\ 0 & \sigma_{cY}^2 \end{bmatrix} \tag{5.15}$$

The rotation matrix R is defined as follows:

$$\begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \tag{5.16}$$

where α is the orientation angle of the OBB.

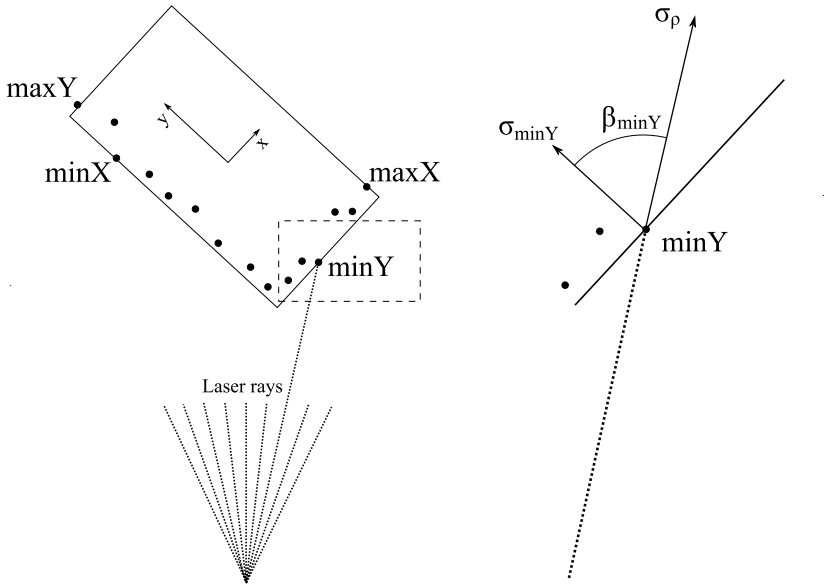

Figure 5.6: Computation of the position uncertainty of an extreme point of the extracted OBB.

The center positions cX and cY become correlated in the ego-vehicle coordinate system. However, we neglect the covariances σ_{cXY} and σ_{cYX}, which are set to 0.

To compute the orientation angle uncertainty σ_θ^2, the edge of the convex-hull, which coincides with the best aligned OBB, is used. The end points of the coinciding edge are saved during the RC algorithm. The end points are input of the weighted total-linear-regression method, incorporating the laser range finder sensor model [5]. The ith raw data point has a radial uncertainty σ_{ρ_i}, while the angular uncertainty is neglected.

$$\sigma_\theta^2 = \frac{1}{(D^2+N^2)^2}\sum (A-B)^2\sigma_{\rho_i}^2 \tag{5.17}$$

where $A = N(\bar{x}\cos\theta_i - \bar{y}\sin\theta_i - \rho_i\cos 2\theta_i)$,
$B = D(\bar{x}\sin\theta_i + \bar{y}\cos\theta_i - \rho_i\sin 2\theta_i)$,
$D = \sum[(\bar{y}-y_i)^2 - (\bar{x}-x_i)^2]$, $N = -2\sum(\bar{y}-y_i)(\bar{x}-x_i)$, $\bar{x} = \frac{1}{N}\sum\rho_i\cos\theta_i$,
$\bar{y} = \frac{1}{N}\sum\rho_i\sin\theta_i$.

The used sensors (real LRF ones as well as simulated ones) have a constant value of σ_{ρ_i} for each laser ray and is equal to $\sigma\rho_i = \sigma_\rho$.

5.1.4 Simulator based results

For the test of the proposed algorithm, a Laser Range Finder (LRF) is mounted in front of the ego-vehicle. The step angle for the LRF is set to 1° with an angle range of 180°. The test scenario consists of a vehicle running according to a circular trajectory in front of the ego-vehicle. To compare the proposed approaches, three attributes of the real object and estimated OBB representation are measured: orientation angle, distance and length of the more visible side (see Figure 5.7). The visible object's side, for which the angle between its normal vector and the line passing by the origin of the coordinate system and the object is the smallest, is called the more visible side. The angle between side's normal vector and the line passing by the origin of the coordinate system and the object is called visibility angle.

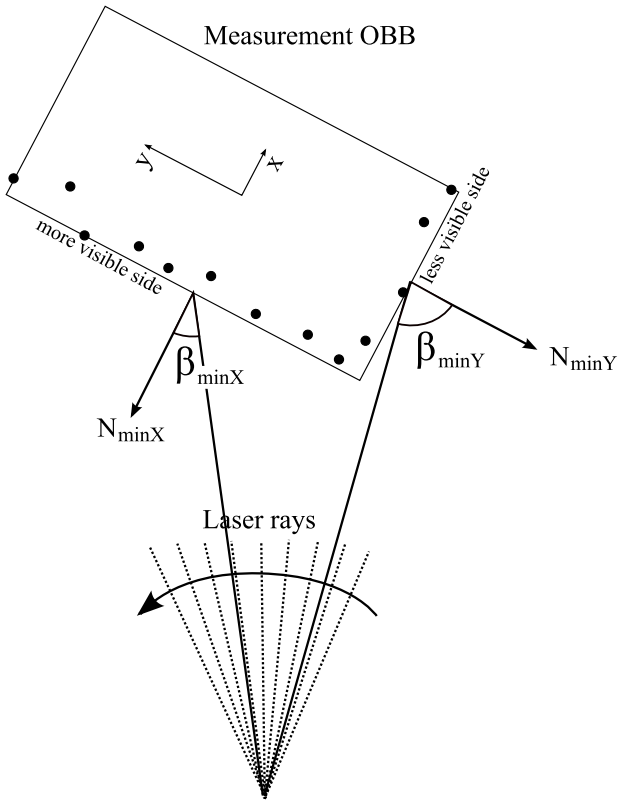

Figure 5.7: More and less visible side of the measurement OBB.

Figures 5.8 - 5.13 show measured values (versus real ones) of the distance, angle and side's length respectively, where the sensor range uncertainty σ_ρ is set to 0.1m. The convex-hull based and the LT based contour finding

methods are used to obtain object OBB representation. In the case of the LT based method, the threshold is set to 1.645.

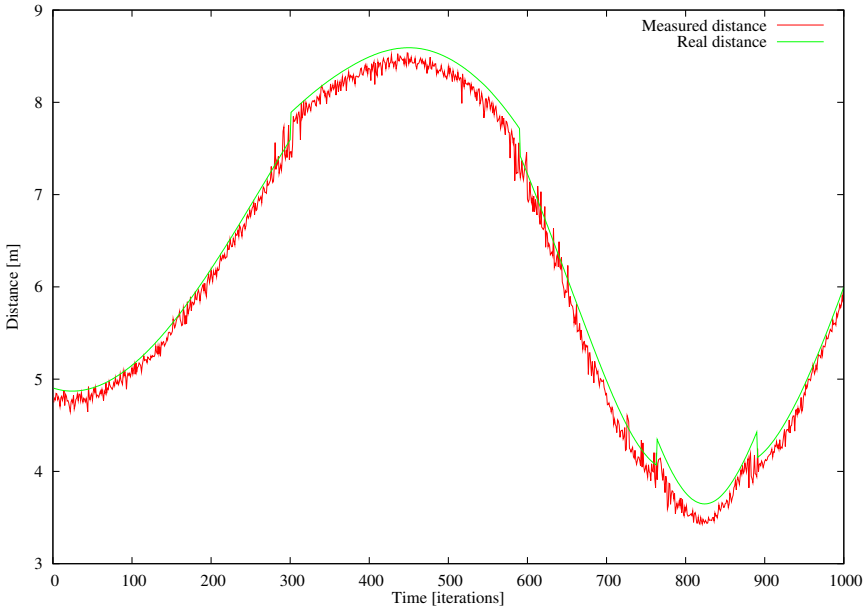

Figure 5.8: Distance of the more visible side (convex-hull).

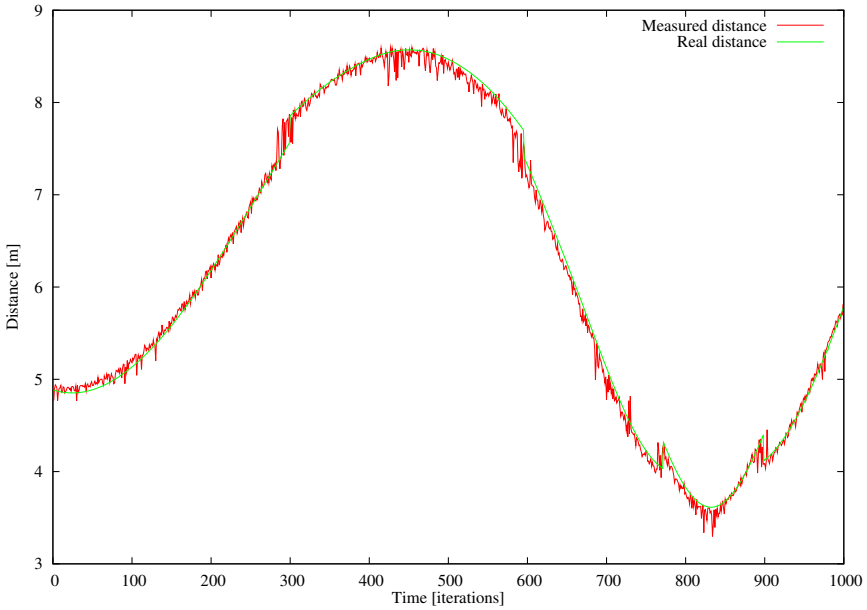

Figure 5.9: Distance of the more visible side (LT, T=1.645).

As a measure of accuracy of the proposed algorithms, averaged absolute error of the measured values, over 1000 time steps, is used. Tables 5.1 - 5.3 show the error of the different methods for different values of the sensor

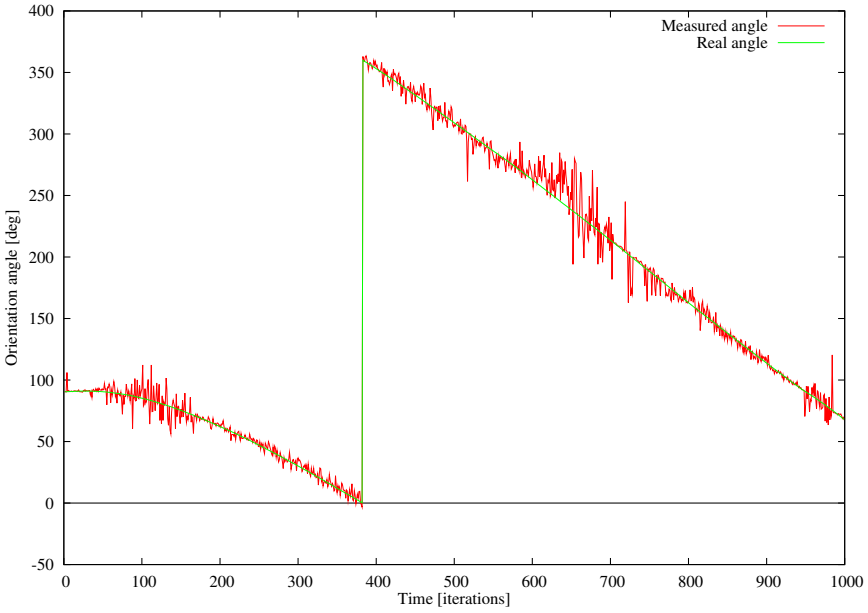

Figure 5.10: Orientation angle (convex-hull).

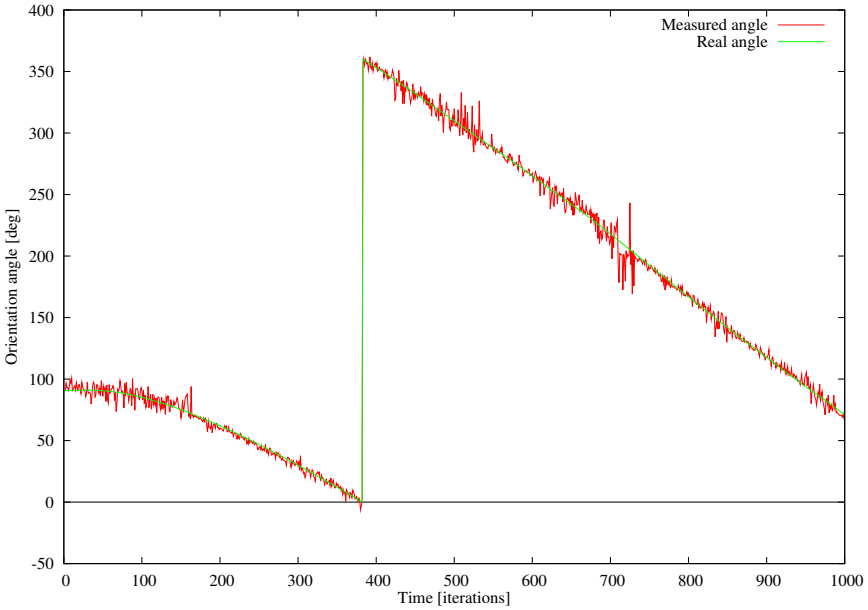

Figure 5.11: Orientation angle (LT, T=1.645).

range uncertainty σ_ρ. T is the threshold used for the LT algorithm. The angle is expressed in degrees and lengths in meters.

One may see (Tables 5.1, 5.2 and 5.3) that none of the methods outperforms significantly the other in terms of accuracy. However, there are cases where the methods perform differently. The LT based method is more accurate in case of distance estimation. This is expected, since LT based technique takes into account all points to create the contour, when the c-Hull

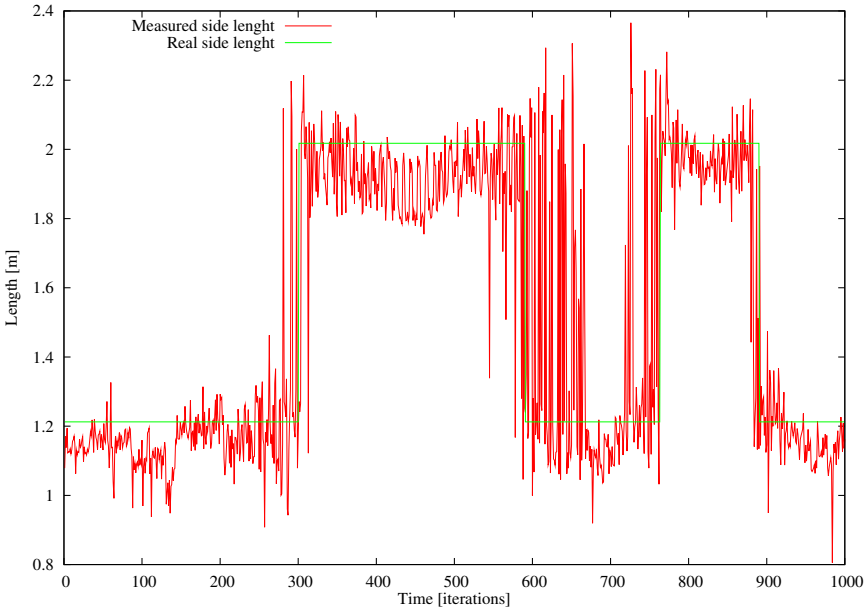

Figure 5.12: The more visible side length (convex-hull).

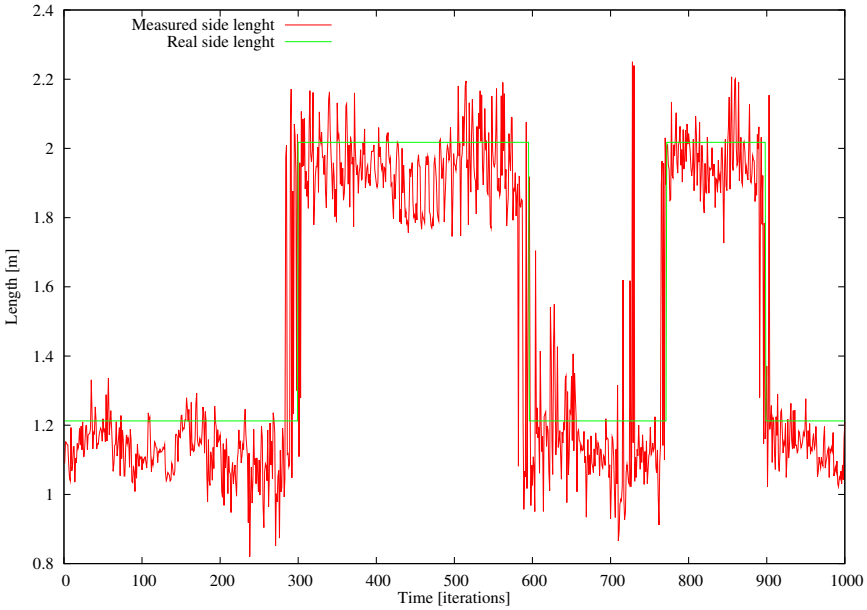

Figure 5.13: The more visible side length (LT, T=1.645).

based method takes only the most external ones. However, in real urban environment conditions, c-Hull distance error can be neglected because its relative value is small. In figure 5.12, between 600 and 700 iteration, one can see that the estimated length value, in some iterations, is greater that the real one. This is due to the fact, that we compare the more visible side length. Even a small difference between estimated and real orientation angle in the moment when the visibility angles of the two sides are similar

Table 5.1: Distance error.

	σ_ρ	LT, T=1.28	LT, T=1.645	LT, T=2.576	convex-hull
1	0.2	0.130	0.157	0.180	0.269
2	0.1	0.081	0.082	0.080	0.138
3	0.01	0.060	0.055	0.065	0.056
4	0.005	0.057	0.056	0.060	0.061

Table 5.2: Angle error.

	σ_ρ	LT, T=1.28	LT, T=1.645	LT, T=2.576	convex-hull
1	0.2	0.177	0.226	0.275	0.176
2	0.1	0.067	0.064	0.089	0.085
3	0.01	0.022	0.015	0.013	0.010
4	0.005	0.013	0.012	0.009	0.007

Table 5.3: The object's side length error.

	σ_ρ	LT, T=1.28	LT, T=1.645	LT, T=2.576	convex-hull
1	0.2	0.216	0.281	0.384	0.195
2	0.1	0.125	0.126	0.152	0.131
3	0.01	0.097	0.095	0.102	0.090
4	0.005	0.099	0.096	0.092	0.094

may case that the different sides are considered to be more visible for extracted OBB and real object. Results show also that the threshold used in the LT based method plays an important role the error decrease for higher range sensor uncertainties. By a good adjustment of the threshold, the LT method can perform similarly as the c-Hull based one.

The other important aspect related to the real-time algorithms is the computation complexity. Let us define s as the number of the line segments and n as the number of the points used to construct the OBB. The on-line convex-hull based contour finding has the optimistic complexity of $O(n)$, and pessimistic complexity of $O(n^2)$. The algorithm of rotating calipers has the complexity of $O(s_c)$, where s_c is the number of the convex-hull segments. The line segment based contour finding has the complexity of $O(sn^2)$.

By taking into consideration the results and real sensor range uncertain-

ties, the c-Hull based contour finding method is more suitable for real-time processing. Furthermore, the c-Hull based method gives estimations with similar error as LT based one with low sensor range uncertainties, corresponding to the most real sensor cases. However, the characteristics of the two methods could be exploited to build an adaptive approach.

5.1.5 Conclusion

An oriented bounding box representation method for tracking objects is presented. Two variants, depending on how objects' contours are determined, are proposed. The first variant is based on a convex-hull contour finding. The second one uses line segments extraction. The two variants are tested and compared in terms of accuracy and complexity. The results show that the two variants behave similarly in terms of accuracy, while the convex-hull based variant is more interesting than the line segments based one considering computation complexity.

5.2 Tracking

The object's state estimation is obtained by the means of Extended Kalman Filter (EKF). All values of the track's state vector are expressed in the local ego-vehicle coordinate system. Tracks are represented by the augmented OBB state vector x_t :

$$x_t = [cx, \dot{cx}, cy, \dot{cy}, \theta, \dot{\theta}, dx, dy]^T \tag{5.18}$$

In the model, the odometry information is taken into account to allow objects tracking from a mobile platform. The state change of the ego-vehicle is represented as differences of position $\Delta x, \Delta y$ and angle $\Delta\gamma$ between consecutive instants. Thus, the input to the state transition equation is defined as:

$$u_k = [\Delta x, \Delta y, \Delta\gamma] \tag{5.19}$$

The Discrete White Noise Acceleration Model (DWNA) [10] is used to describe objects kinematics and process noise. Thus, taking into account the odometry information, the track state transition is modelled as follows:

$$x_{t|t-1} = A(\Delta x, \Delta y, \Delta\gamma)Fx_{t-1} + Bu_t + Gv_{t-1} \tag{5.20}$$

where A is defined as follows:

$$A(\Delta x, \Delta y, \Delta\gamma) = \begin{bmatrix} c & 0 & s & 0 & 0 & 0 & 0 & 0 \\ 0 & c & 0 & s & 0 & 0 & 0 & 0 \\ -s & 0 & c & 0 & 0 & 0 & 0 & 0 \\ 0 & -s & 0 & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.21}$$

The prefixes s and c abbreviate the $\sin(\Delta\gamma)$ and $\cos(\Delta\gamma)$.

F is the standard DWNA transition matrix, expressed as follows:

$$F = \begin{bmatrix} 1 & t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.22}$$

t is the length of the sampling period,

B is the odometry-input model, expressed as follows:

$$B = \begin{bmatrix} -\cos(\Delta\gamma) & \sin(\Delta\gamma) & 0 \\ 0 & 0 & 0 \\ \sin(\Delta\gamma) & -\cos(\Delta\gamma) & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.23}$$

G represents the noise gain matrix, which is expressed as follows:

$$G = \begin{bmatrix} \frac{1}{2}t^2 & 0 & 0 & 0 & 0 \\ t & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}t^2 & 0 & 0 & 0 \\ 0 & t & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2}t^2 & 0 & 0 \\ 0 & 0 & t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.24}$$

According to the chosen kinematics model, the process noise is defined with the Gaussian distribution:

$$v_{t-1} = [\ddot{cx}, \ddot{cy}, \ddot{\theta}, \hat{\sigma}_{\bar{dx}}, \hat{\sigma}_{\bar{dy}}], \quad v_{t-1} \sim N(0, Q) \tag{5.25}$$

where

$$Q = G v_{t-1} G^T \tag{5.26}$$

with $\hat{\sigma}_{\bar{dx}}$ and $\hat{\sigma}_{\bar{dy}}$ are the process errors for the OBB size dx and dy respectively.

The predicted estimation covariance matrix is :

$$P_{t|t-1} = \frac{\partial A}{\partial x}(x_{t-1}) F P_{t-1} \frac{\partial A^T}{\partial x}(x_{t-1}) F^T + Q_t \tag{5.27}$$

The observation equation can be written as follows:

$$z_t = H x_{t|t-1} + w_t \tag{5.28}$$

where H is the observation model, expressed as:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.29}$$

w_t is defined with a Gaussian distribution:

$$\begin{gathered} w_t \sim N(0, R), \\ R = \sigma_z^2 I_{5,5} \end{gathered} \tag{5.30}$$

where $I_{5,5}$ is the identity matrix.

5.2.1 Size change compensation in velocity estimation

During the tracking, the detected size of the object changes. This is due to the detection of a new part of the object, or due to the disappearance of already detected part. The size change can introduce on the object a "phantom" movement (see Figure 5.14), which influences velocity estimation. This is due to the fact that the object reference point is the center of the OBB. To avoid this effect, the size change compensation must be applied.

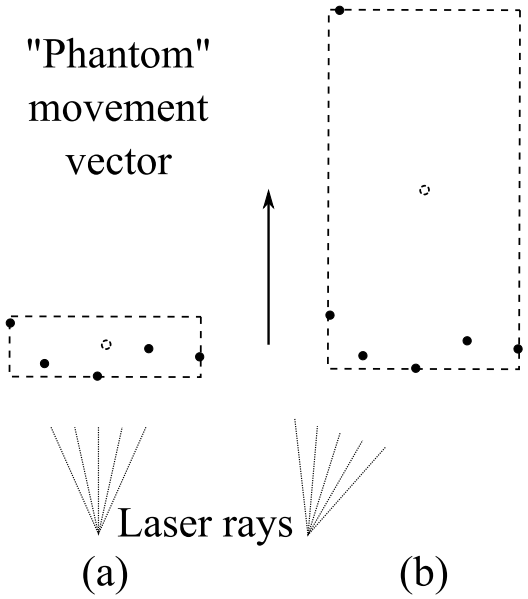

Figure 5.14: "Phantom" movement vector.

The algorithm starts by computing a measurement size change vector Δd_{meas} which corresponds to the difference between the measurement size and the predicted size:

$$\begin{aligned} \Delta d_{meas}[x] &= \tfrac{1}{2} z_{[}dx] - \hat{x}(t|t-1)[dx] \\ \Delta d_{meas}[y] &= \tfrac{1}{2} z_{[}dy] - \hat{x}(t|t-1)[dy] \end{aligned} \tag{5.31}$$

After transformation, this vector is used to translate the measurement before filtering correction stage takes place.

Before the transformation, the direction factor is computed. It expresses the direction of the size change and the relative position of the tracked object to the sensor.

$$\begin{aligned} dir_x &= -sign(d_{meas}[x])sign(\hat{x}(t|t-1)[cx]) \\ dir_y &= -sign(d_{meas}[y])sign(\hat{x}(t|t-1)[cy]) \end{aligned} \tag{5.32}$$

The measurement size change vector Δd_{meas} is expressed in the OBB local coordinate system, and thus, it must be rotated to obtain a translation vector Δd^{ego}_{meas} in the ego-vehicle coordinate F system.

In the rotation equation, the direction of the direction factor is taken into account:

$$\begin{aligned} \Delta d^{ego}_{meas}[x] &= dir_x|\cos(\alpha)\Delta d_{meas}[x]| + dir_y|\sin(\alpha)\Delta d_{meas}[y]| \\ \Delta d^{ego}_{meas}[y] &= dir_x|\sin(\alpha)\Delta d_{meas}[x]| + dir_y|\cos(\alpha)\Delta d_{meas}[y]| \end{aligned} \tag{5.33}$$

where α is the measurement OBB orientation angle.

The obtained vector is then used to translate the measurement centre:

$$\begin{aligned} z[cx] &= z[cx] + \Delta d^{ego}_{meas}[x] \\ z[cy] &= z[cy] + \Delta d^{ego}_{meas}[y] \end{aligned} \tag{5.34}$$

The translated measurement is used in the estimation correction stage of filtering, while the measurement is translated back to its previous position

$$\begin{aligned} z[cx] &= z[cx] - \Delta d_{ego}[x] \\ z[cy] &= z[cy] - \Delta d_{ego}[y] \end{aligned} \tag{5.35}$$

After the correction step of the filtering, the center of the track is ill-positioned, and it should be translated. The size of the track undergoes the process of filtering, and because of this, the magnitude of the track size change is not the same as the magnitude of the measurement size change vector. To translate the track, an estimation size change vector d_{est} is found:

$$\begin{aligned} \Delta d_{est}[x] &= \tfrac{1}{2}\hat{x}(t|t)[dx] - \hat{x}(t|t-1)[dx] \\ \Delta d_{est}[y] &= \tfrac{1}{2}\hat{x}(t|t)[dx] - \hat{x}(t|t-1)[dy] \end{aligned} \tag{5.36}$$

The vector is then rotated to obtain the translation vector, expressed in the ego-vehicle d^{ego}_{est} coordinate system.

$$\begin{aligned} \Delta d^{ego}_{est}[x] &= -dir_x|\cos(\alpha)\Delta d_{est}[x]| - dir_y|\sin(\alpha)\Delta d_{est}[y]| \\ \Delta d^{ego}_{est}[y] &= -dir_x|\sin(\alpha)\Delta d_{est}[x]| - dir_y|\cos(\alpha)\Delta d_{est}[y]| \end{aligned} \tag{5.37}$$

where α is the track orientation angle. In the rotation equation, the inverse values of the direction factors, computed for the measurement size change, are used.

Finally, the track is translated:

$$\begin{aligned} z[cx] &= z[cx] - \Delta d^{ego}_{est}[x] \\ z[cy] &= z[cy] - \Delta d^{ego}_{est}[y] \end{aligned} \tag{5.38}$$

5.3 Inter-rays

5.3.1 Side visibility and visibility factor

The information about visibility of the OBB sides is used an Inter-Rays algorithm. Thus, let us introduce the notion of the side visibility before describing the Inter-Rays algorithm.

The simple method to decide which side of the two sides along the OBB local axis is visible is to compare their visibility angles to right angle. The side visibility angle is an angle between the OBB's side normal and its radius vector. In Figure 5.15, there are examples of visibility angles for $minX$ and $maxX$ sides. A side for which the visibility angle is smaller than the right angle is considered to be visible. Since the sides are parallel, the second side of the same axis is invisible.

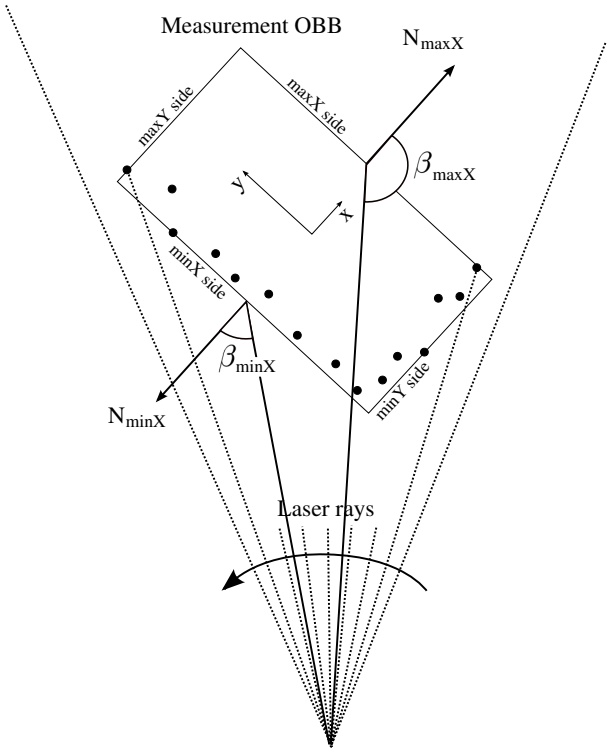

Figure 5.15: Visibility angles associated with maxX and minX sides of the OBB.

Another way of determining side visibilities is to compute visibility factors VF_x and VF_y for the OBB local X axis and local Y axis. The two approaches are proposed.

The first one is described by the equation:

$$VF_x = \frac{max(\beta^f_{minX}, \beta^f_{maxX})}{\beta^f_{minX} + \beta^f_{maxX}} \tag{5.39}$$

where β_{minX} and β_{maxX} correspond respectively to the angles between OBB's sides $minX side$ and $maxX side$ normals and their radius vectors (see Figure 5.15). f is a smoothing parameter, which is set experimentally to 4.

The visibility factor becomes less sensible to the angle difference as the smoothing parameter value increases. The disadvantage of this approach is that, to compute a visibility factor, the two sides are taken into account for each local axis. Since the two sides are parallel, the size and position of the object can influence greatly the difference between the two side visibility angles, and thus, influence the value of the visibility factor.

In the second approach, only one side visibility angle, for each local axis, is used. Considering the OBB local X axis, the smaller angle between the two visibility angles β_{minX} and β_{maxX} is chosen: $\beta_X = min(\beta_{minX}, \beta_{maxX})$. Than, having the angle β_X, the visibility factor for axis X is defined as follows:

$$VF_x = \begin{cases} 1, & \text{if } \beta_X < \beta_T \quad (5.40a) \\ f(x), & \text{if } \beta_T < \beta_X < 90^\circ \quad (5.40b) \\ 0, & \beta_X > 90^\circ \quad (5.40c) \end{cases}$$

where $f(x) = 1 - a^{-(\beta_X - 90^\circ)}$, $a = b^{\frac{1}{90^\circ - \beta_T}}$ (see Figure 5.16). β_T is the maximal visibility angle for which a side is considered as completely visible. It means that if, for a side, $\beta_X \leq \beta_T$, than this side is considered as completely visible. In our experimentations, $b = 0.01$ and $\beta_T = 60^\circ$.

The same principle is considered to compute the visibility factor VF_y for the OBB local Y axis.

The advantage of this visibility factor function is that it takes into account only one side orientation. Thus, the size of the object and the OBB relative position do not influence this factor. In all computations implying the visibility factor, the second approach is used.

5.3.2 Inter-rays algorithm

An important aspect of the OBB extraction is the fact that the raw data points representing the extremities of the extracted OBB do not coincide with the real object's extremities (see Figure 5.17). This aspect allows to provide better object's size and position estimation.

In the Figure 5.17, $minX$, $minY$, $maxX$, $maxY$ are respectively the minimum x coordinate, minimum y coordinate, maximum x coordinate and maximum y coordinate of the extracted OBB. The line Lr (respectively $Lr+n$) is crossing the point $maxX$ (respectively $maxY$) and is perpendicular to the OBB side to which $maxX$ (respectively $maxY$) belongs. The Inter-Rays (IR) real object's extremities position estimation and their variances

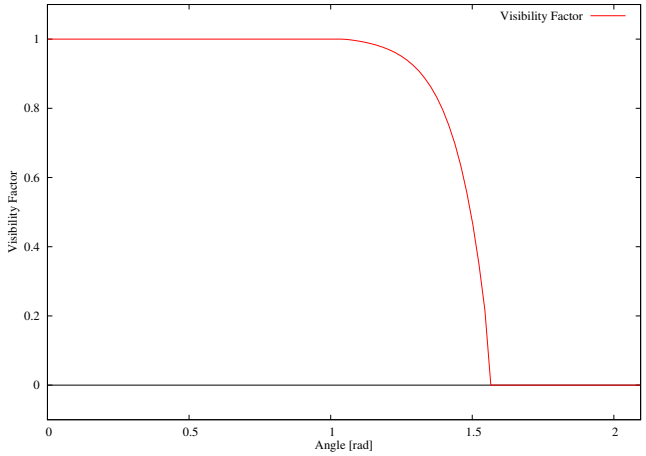

Figure 5.16: Visibility factor function for $b = 0.01$ and $\beta_T = 60°$.

are added to the OBB's size and OBB's size uncertainty. The real object's extremities are situated between the raw data points delimiting the OBB ($maxX$, $maxY$) and the points Pr and $Pr + n$. Pr (respecitvely $Pr + n$) is the intersection point between the ray r (respecitvely $r + n$) with the line Lr (respectively $Lr + n$).

Considering the OBB's local X axis, the real object's extremity position is uniformly distributed with the mean μ_{IRx}, which is equal to the half of the IR line segment length d_{IRx}. The IR line segment is defined by the point $maxX$ and Pr. To fulfil Kalman Filter assumption, the distribution of the real object's extremity position is approximated by a normal distribution with the mean μ_{IRx}, and the variance σ^2_{IRx}, which is set to $(\frac{d_{IRx}}{N_\sigma})^2$. N_σ is the number of sigmas and represents the confidence interval of the approximated distribution, which is equal to the IR line segment length d_{IRx} (see Figure 5.18).

Since the IR values reflect the real object extremity position uncertainty, it should be calculated only for invisible sides. The visible sides are detected by many laser rays, and thus, it is useless to estimate the sides extremity position by Inter-Rays distance. Referring to the Figure 5.17, the IR values should be calculated for sides $maxX$ and $maxY$ points.

The IR line segment length d_{IRx} is computed differently for visible and invisible sides of the local X axis. There are two approaches. The first one, which is the simple is to compute the IR segment length only for invisible sides along the OBB local axis. This approach has, however, a drawback. In the cases, where the visibility angle of the visible side is close to the right angle, the IR values are not calculated. This is not correct since this side is not detected by enough laser rays using only data points to estimate

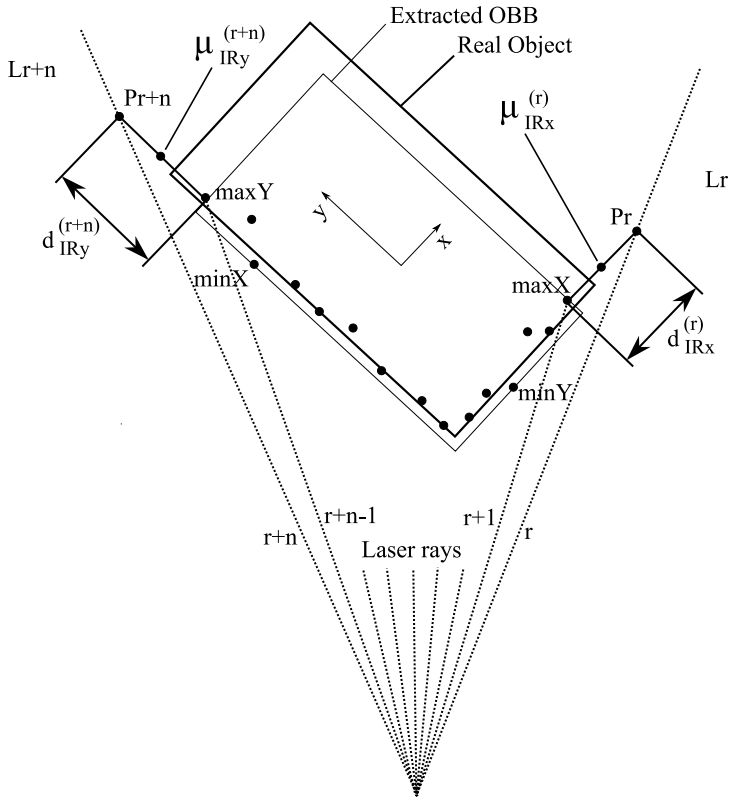

Figure 5.17: Inter-Rays uncertainty paradigm.

its extremity position. In the second approach the IR values are computed relatively to a visibility factor VF_x. The IR segment lengths d_{IRx}^{vis} and d_{IRx}^{inv} are computed for respectively visible and invisible sides. The d_{IRx}^{vis} for visible side is than changed according to the visibility factor:

$$d_{IRx}^{vis} = d_{IRx}^{vis}(1 - VF_x) \tag{5.41}$$

Finally, the d_{IRx} value for the local X axis is a sum of sides IR line segment lengths for visible and invisible sides:

$$d_{IRx} = d_{IRx}^{vis} + d_{IRx}^{inv} \tag{5.42}$$

.

The measurement Inter-Rays values $z[\mu_{IRx}]$ and $z[\sigma_{IRx}^2]$ are used in each iteration of the tracking algorithm to correct the size of the OBB measurement. We introduce two measurements: perceived and corrected. As the name suggests, the perceived measurement holds perceived OBB parameters: centre position, size, orientation, Inter-Rays mean and their uncertainties. The size and position uncertainties stored in the perceived OBB do not

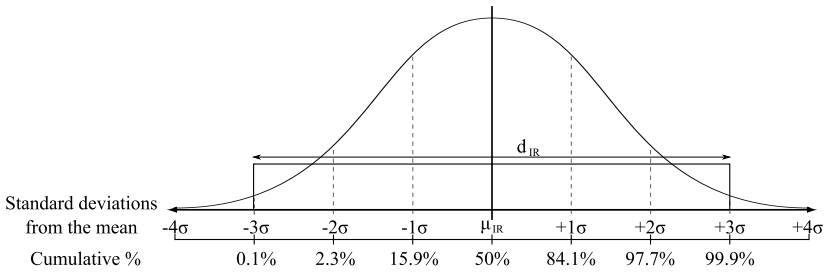

Figure 5.18: Normal distribution approximation.

include the Inter-Rays uncertainty. The corrected measurement consists of corrected values and uncertainties (for example centre position, size). The introduction of the two types of measurements allows comparison of different algorithms since the perceived OBB stays unchanged during one iteration of the tracking algorithm. The other advantage is that it structures the algorithms, and thus, provides better insight into them.

Before correction stage takes place, all data from the perceived OBB is copied to the corrected measurement. Then, some of the OBB parameters are corrected.

The correction equations are expressed as follows:

$$z[dx] = z_{perc}[dx] + z_{perc}[\mu_{IRx}] \tag{5.43}$$

$$z[\sigma^2_{dx}] = z_{perc}[\sigma^2_{dx}] + z_{perc}[\sigma^2_{IRx}] \tag{5.44}$$

where z_{perc} is the perceived measurement, z is the corrected measurement used in the track state estimation process.

The measurement center position uncertainty is computed in the same manner as before (see Section 5.1.3). The obtained center position uncertainty is greater than the uncertainty given in the case of the OBB extracted without applying the Inter-Rays algorithm. It, however, reflects better the measurement position uncertainty.

The same process is applied for the OBB's local Y axis.

In certain raw data point configurations, it happens that the IR line segment lengths d_{IRx} and d_{IRy} are large. It may cause great overestimation and tracking instability. To avoid this situation, the IR line segment lengths d_{IRx} and d_{IRy} are limited to a certain values $T_{d_{IRx}}$ and $T_{d_{IRy}}$.

5.3.3 Centre translation

After correcting the measurement's size, the measurement's centre must be translated. This is due to the fact that, after size change, the corrected measurement sides change their position. The new sides positions do not correspond to the tracked object surfaces (see Figure 5.19(a)). The introduction of the center translation solves this problem (see Figure 5.19(b)).

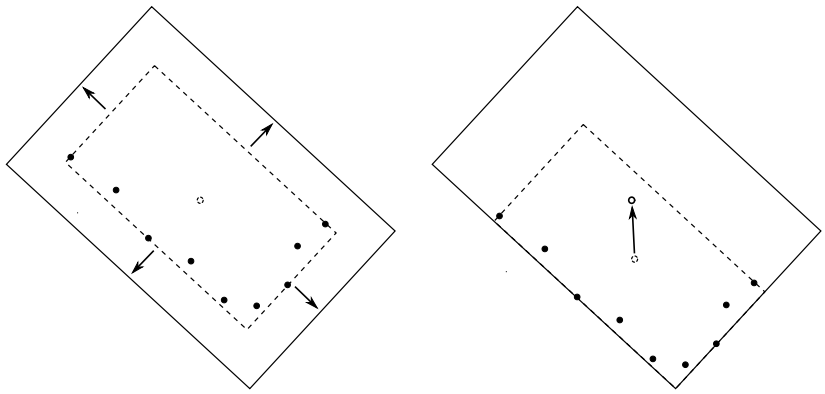

Figure 5.19: Center translation after OBB size change: (a) size change (b) center translation.

The centre translation vector Δ_c is proportional to the size change vector Δ_d. In the case of the IR algorithm, the size change vector Δ_d is as follows:

$$\Delta_d = (z_{perc}[\mu_{IRx}], z_{perc}[\mu_{IRy}]) \tag{5.45}$$

There are two different approaches, which define the relations between the two vectors.

The first approach is a simple centre translation, in which the relation is defined as follows:

$$\Delta_c = \frac{1}{2}(DF_x \cdot \Delta_{d_x}, DF_y \cdot \Delta_{d_y}) \tag{5.46}$$

where DF_x and DF_y are the direction factors:

$$DF_x = \begin{cases} 1, & \text{if } \beta_{minX} < 90^\circ \quad (5.47a) \\ -1, & \text{if } \beta_{maxX} < 90^\circ \quad (5.47b) \\ 0, & \text{otherwise} \quad (5.47c) \end{cases}$$

The same expression is used to define DF_y by considering the OBB local Y axis.

The direction factor allows to translate the centre towards the invisible sides, for which the IR values are computed. In the case where two parallel sides are classified as invisible, their visibility angles are greater than 90° (see section 5.3.1). In this case, there is no translation along the corresponding OBB local axis. Thus, the direction factor is set to 0.

The second approach uses the visibility factor to calculate the translation vector. In this approach, the translation of the centre is relative to the visibility of the OBB sides. The visibility factor permits to compute the centre translation coefficient, which is proportional to the visibility of the sides. The centre translation vector is computed according to the equation below:

$$\Delta_c = \frac{1}{2}(VF_x \cdot DF_x \cdot \Delta_{d_x}, VF_y \cdot DF_y \cdot \Delta_{d_y}) \tag{5.48}$$

The introduction of the visibility factor based translation reduces the effect of "phantom" object movement. The effect is a quick object position change, which can affect object's velocity estimation since a filter can smooth these fluctuations only to some extend.

This effect occurs when a transition between the stage with only one side, classified as visible, and the stage where two sides are classified as visible (and reversely), takes place. Let us use Figure 5.20 to illustrate the problem. At the first instant t, only one side is considered as visible. Due to the rotation of the object, in the second instant $t + 1$, there are two sides detected as visible. In the figure, one can see that the corrected OBB has changed its position rapidly. The tracked object, in the example, has no linear velocity. However, when a simple centre translation is used, the linear velocity estimation tends to be visibly greater than the real one. The greater is the IR mean, the greater the "phantom" object movement is. So, the effect becomes more prominent with the increase of the objects distance, and thus, with the increase of the angular resolution of the sensor.

There is another important point. Due to the transitions, the size of the extracted OBB changes. This also can cause the effect of "phantom" object movement. This problem, however, is covered in section 5.2.1.

To express the centre in the ego-vehicle coordinate system, the last stage of the OBB centre translation is the rotation of the translation vector by the OBB orientation angle α:

$$\Delta_c = R * \Delta_c \tag{5.49}$$

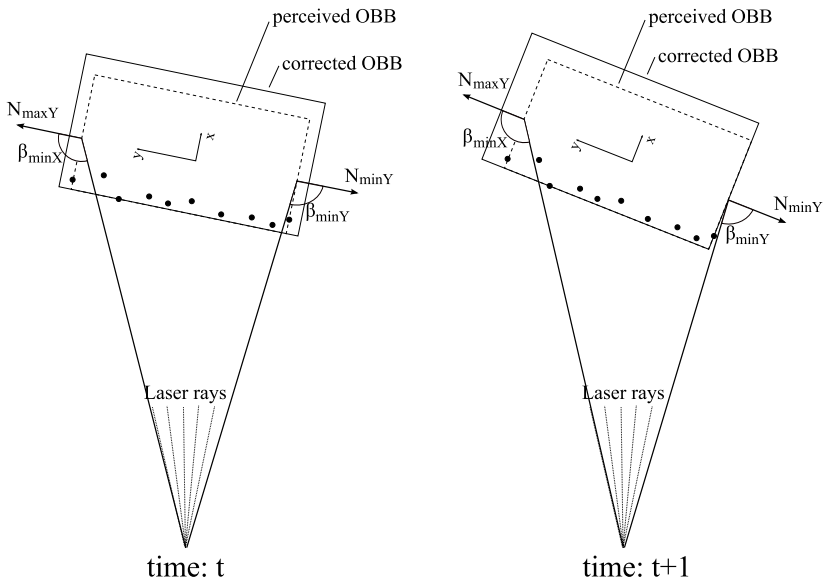

Figure 5.20: Simple centre translation.

where R is a rotation matrix defined as follows:

$$\begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \tag{5.50}$$

5.4 Fixed Size

The idea of the fixed size (FS) assumption is based on the fact that, in general cases, objects' size does not change during the tracking. However, due to the LRF's limited resolution and change of the relative distance and orientation of the observed object, measurements of the object's size vary in time. The principle of the FS assumption is that the size of the track representing the tracked object can change depending on the IR uncertainty. The FS algorithm takes place in each iteration of the tracking after the track prediction and measurements extraction, and allows to keep the best object's size estimation obtained up to the current iteration.

For the following algorithm description, we consider the local OBB's X axis. The same process is applied to the local OBB's Y axis.

We assume that the OBB measurement, with IR values, are available.

The first step of the algorithm is to find a perceived size of the corrected measurement. The perceived size of the corrected measurement is set to:

$$z[dx] = min(z_{perc}[dx], \hat{x}_{perc}(t|t-1)[dx]) \tag{5.51}$$

where $z_{perc}[dx]$ is the size perceived in the current iteration. $\hat{x}(t|t-1)[dx]$ is the maximum perceived size up to the current iteration. It can be found by the following equation:

$$\hat{x}_{perc}(t|t-1)[dx] = \hat{x}(t-1|t-1)[dx] - \hat{x}(t-1|t-1)[\mu_{IRx}] \tag{5.52}$$

The second step consists of finding the IR mean for the corrected measurement. The general rule is to use the smallest IR mean observed up to the current iteration. There are, however, cases when the currently observed IR mean should be used. Indeed, when the perspective changes and the perceived size of the object changes by a great amount, the IR mean observed up to the current iteration becomes invalid, and should not be used to compute the corrected measurement IR values. To detect this situation let us define the perceived size difference:

$$\Delta dx_{perc} = z_{perc}[dx] - \hat{x}_{perc}(t|t-1)[dx]) \tag{5.53}$$

Then, the perceived size difference Δdx_{perc} is compared to the smallest IR distance observed up to the current iteration:

$$\hat{x}(t-1|t-1)[d_{IRx}] = 2\hat{x}(t-1|t-1)[\mu_{IRx}] \tag{5.54}$$

If $\Delta dx_{perc} > \hat{x}(t-1|t-1)[\mu_{dIRx}]$, then the corrected measurement IR mean is set to the perceived IR mean:

$$z[\mu_{IRx}] = z_{perc}[\mu_{IRx}] \tag{5.55}$$

else, the corrected measurement IR mean is set to smaller of the perceived IR mean and the smallest IR mean observed up to the current iteration:

$$z[\mu_{dIRx}] = min(z_{perc}[\mu_{dIRx}], \hat{x}(t-1|t-1)[\mu_{dIRx}]) \tag{5.56}$$

The obtained corrected measurement IR mean is stored in the track:

$$\hat{x}(t-1|t-1)[\mu_{dIRx}] = z[\mu_{dIRx}] \tag{5.57}$$

Next, the correction stage of IR algorithm takes place (5.43) and (5.44).

Like in the case of the IR algorithm, the size of the measurement OBB changes, and thus, a centre translation must be applied (see section 5.3.3). The same centre translation algorithm as in the case of the IR method is used. The size change vector, however, is different and is defined as follows:

$$\Delta_d = (z[dx] - z_{perc}[dx], z[dy] - z_{perc}[dy]) \tag{5.58}$$

5.5 Results

5.5.1 Simulator based results

To show the interest of using the IR uncertainty and the FS assumption, a scenario where a tracked vehicle runs in a circle trajectory in front of the instrumented vehicle is installed. This scenario is chosen because it integrated difficult situations during the tracking process. Indeed, the tracked object changes its orientation all of the time as well as its linear velocity.

In the graphs presented in this subsection, there are results of four algorithms for the same scenario. All of the compared algorithms are based on Extended Kalman Filter. The first one, called "EKF-noComp" consists of tracking without the application of the size change compensation. The second "EKF" consists of tracking with the size change compensation. The third algorithm, called "IR" is the previous algorithm with the Inter-Rays uncertainty. Finally the fourth one, called "FS" consists of tracking with the Fixed Size assumption. The X axis the in figures represents the number of iterations of the tracking algorithm. The frequency of the sensor is set to 50 Hz, thus one iteration takes 0.1 seconds. This value can be tuned in the simulator. All the results presented in the graphs are expressed in the instrumented vehicle local coordinate system.

Figure 5.21 shows the estimated trajectories and the real one of the tracked vehicle in the tested scenario. From the figure, one can see that the FS algorithm outperforms the others in terms of position estimation precision. The position estimation will be further discussed in more details.

The results of angle orientation and its velocity are presented in figures 5.22-5.27. The stage of angle orientation and angular velocity estimation is identical for all algorithm discussed in this chapter and thus, only a measurement an estimation of the angle and the real value are shown. The first part, between the beginning and around iteration 130, corresponds to the moment when only one side of the vehicle is visible. One can see in Figure 5.23 that the standard deviation of the measurement and estimation in the first part is greater than in the other moments of the tracking. This is related to the length of the convex-hull segment, found in the calipers algorithm, which is used to compute the orientation uncertainty. The shortest the segment the more uncertain the orientation angle becomes.

In figure 5.25, presenting angular velocity, one can see that the estimation "follows" the real value, but never converges with it. This fact can be explained by the usage of the Discrete White Noise Acceleration kinematic

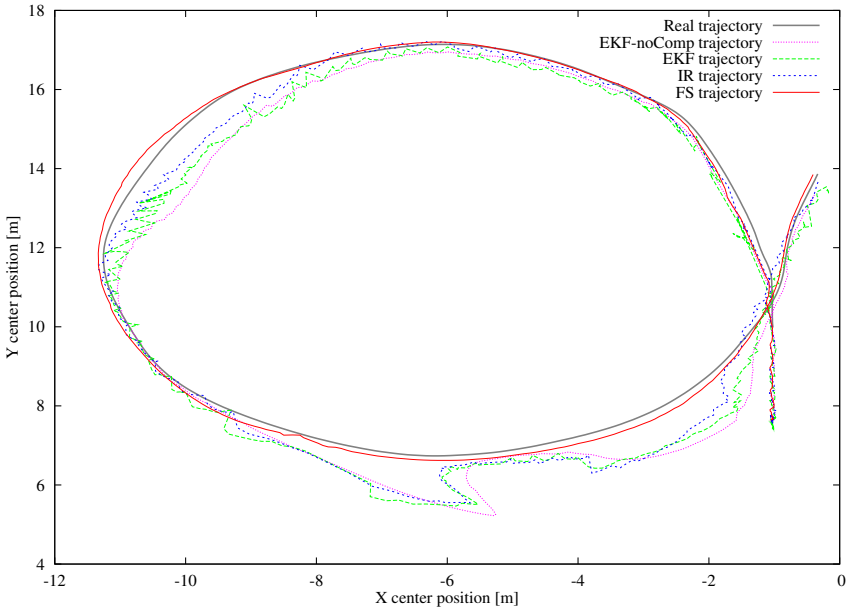

Figure 5.21: Trajectory of the tracked vehicles.

model (DWNA), in which acceleration is not estimated. This implies that in the system model, a transition between the states is done with preserving the velocity from the previous time. In figure 5.25 one can see that the real angular velocity is changing all the time, and thus, the estimated value cannot converge to the real value.

The IR and FS algorithms influence directly the estimation of the tracked object size. The size estimation results are shown in figures 5.28-5.35. In figure 5.28 and 5.31, one can see FS based size estimation is stable during the tracking. The estimation based on the other algorithms undergoes oscillations. This is due to the continuous change of detection points configuration. Another aspect is accuracy of the object's size estimation. One can see in figures 5.30 and 5.33, representing the size estimation error, that the FS method gives more accurate results that the other methods. The places when the estimations present great deviation from the real object size, correspond to the situations when only one side of the object can be perceived by the sensor. Thus, only one object's dimension information is available. One can see that the usage of the FS assumption, which stores the object size, allows to obtain reliable estimation, even in the cases when only one object's side is seen. Figures 5.29 and 5.32 present the size estimation standard deviation. The IR and FS based values are greater than values obtained without IR and FS. This is a result of adding IR uncertainty.

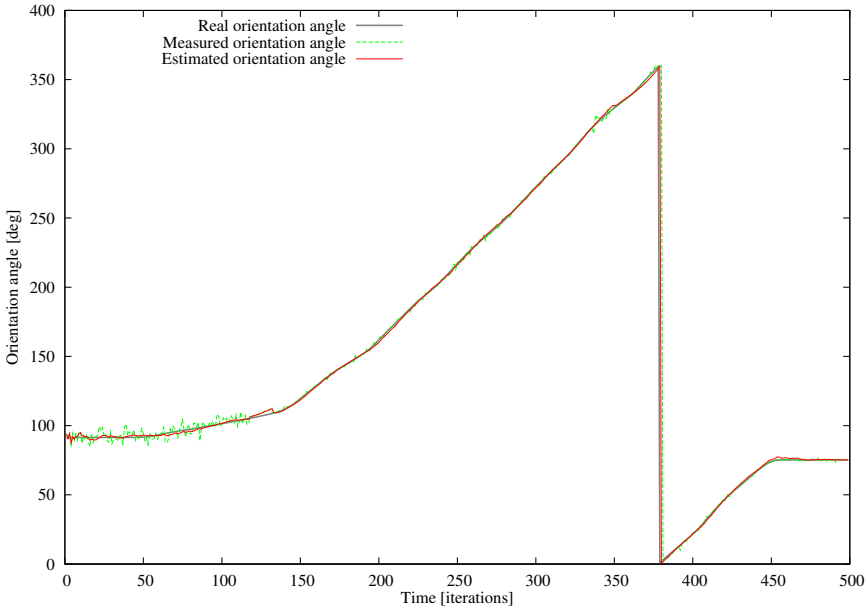

Figure 5.22: Orientation angle estimation.

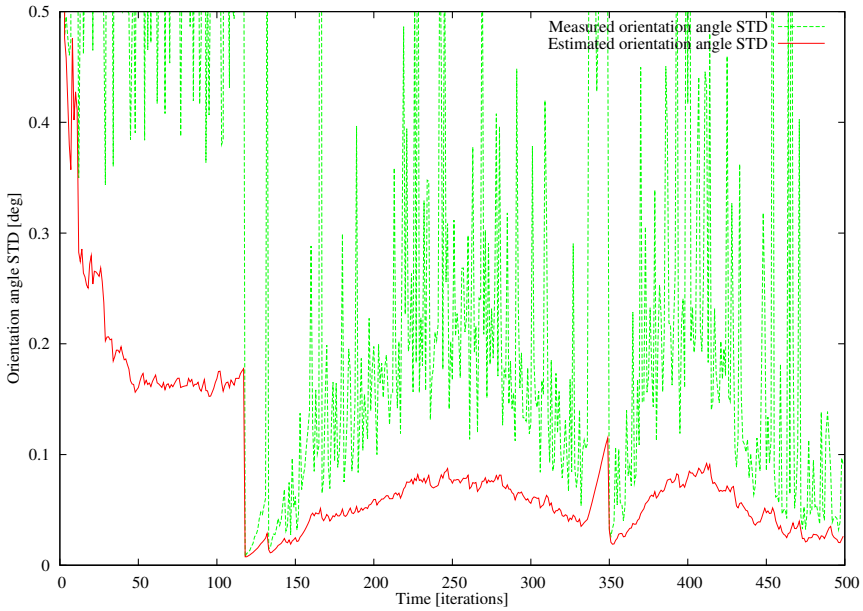

Figure 5.23: Orientation angle standard deviation.

However, storing the smallest perceived IR mean in the FS approach (see Figures 5.34 and 5.35) leads to much smaller side estimation standard deviation when compared with the IR algorithm case.

The results of position estimation can be seen in figures 5.36 - 5.38. Figures 5.36 and 5.39 show the estimated and real object positions in X

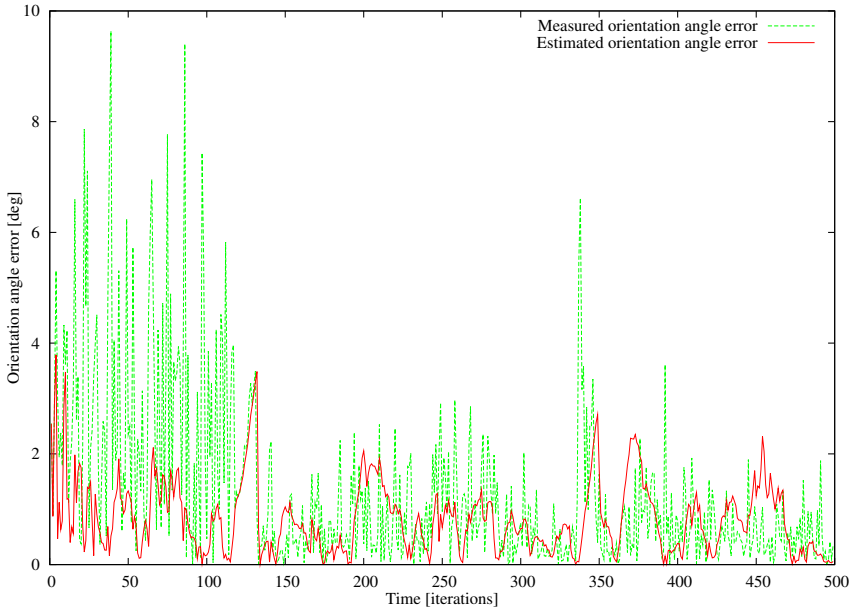

Figure 5.24: Difference between real and estimated orientation angles.

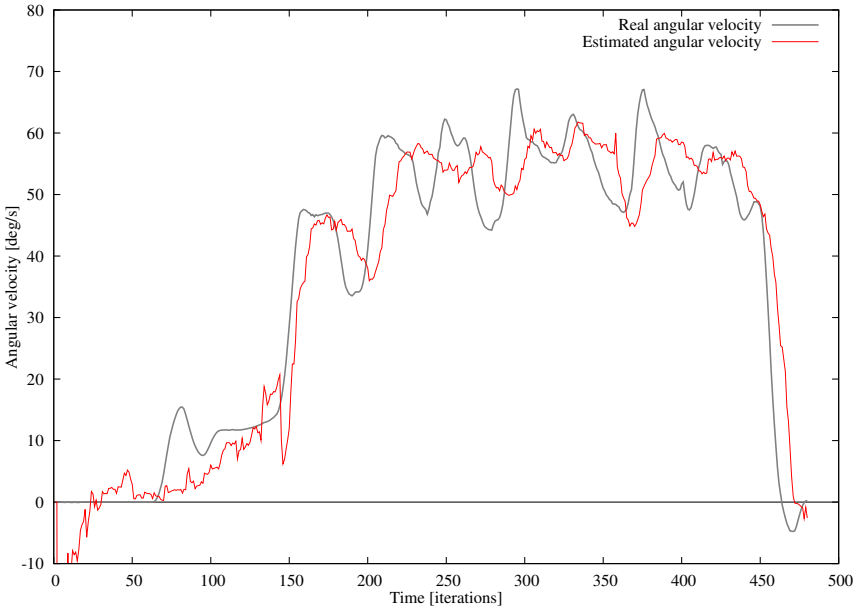

Figure 5.25: Angular velocity estimation.

and Y coordinate respectively. Figures 5.38 and 5.41 show the difference between the estimated position and the real one in X and Y coordinate respectively.

One can see in the mentioned figures that the FS approach outperforms the other in terms of object's center position estimation. The better center position estimation can be obtained thanks to the more reliable object's size estimation. This is due to the proportional relation between the size and

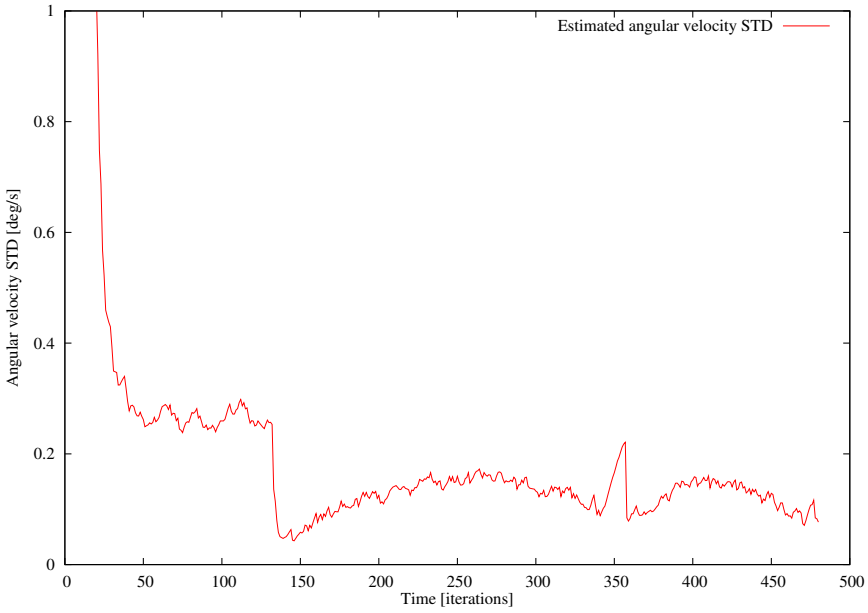

Figure 5.26: Angular velocity covariance.

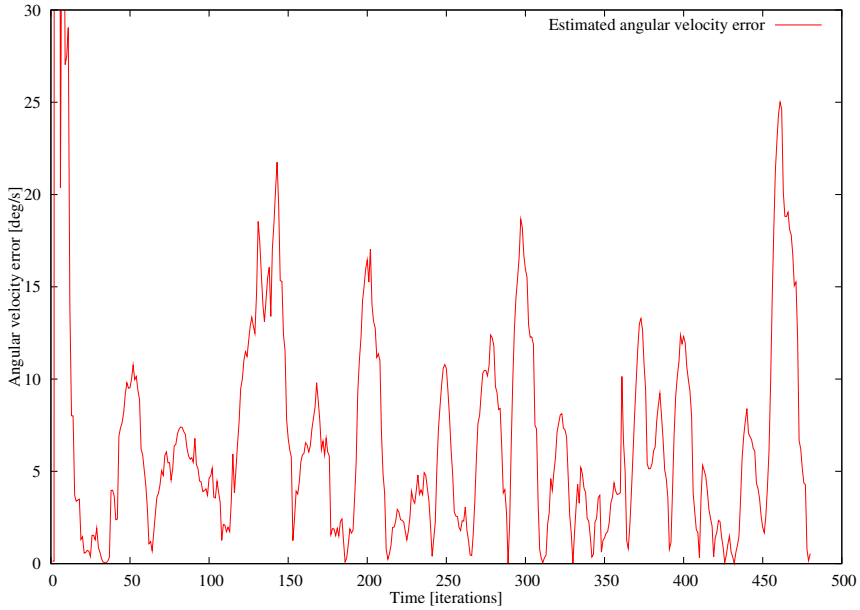

Figure 5.27: Difference between real and estimated angular velocities.

the center of the OBB. Figures 5.37 and 5.40 show the standard deviation (STD) of the position estimation in X and Y coordinate respectively. One can see in the mentioned figures that STD values of the IR and FS are greater than in the other cases. This is due to the introduction of the IR uncertainty.

Figures 5.42-5.47 show the velocity estimation results. In figures 5.42 and 5.45 the estimated and real object velocities in X and Y coordinate

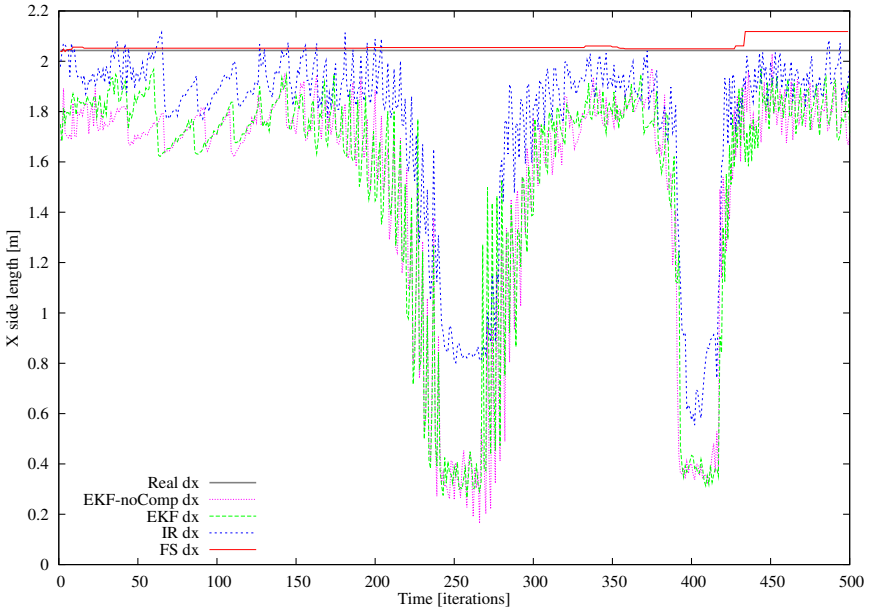

Figure 5.28: Size estimation (X coordinate).

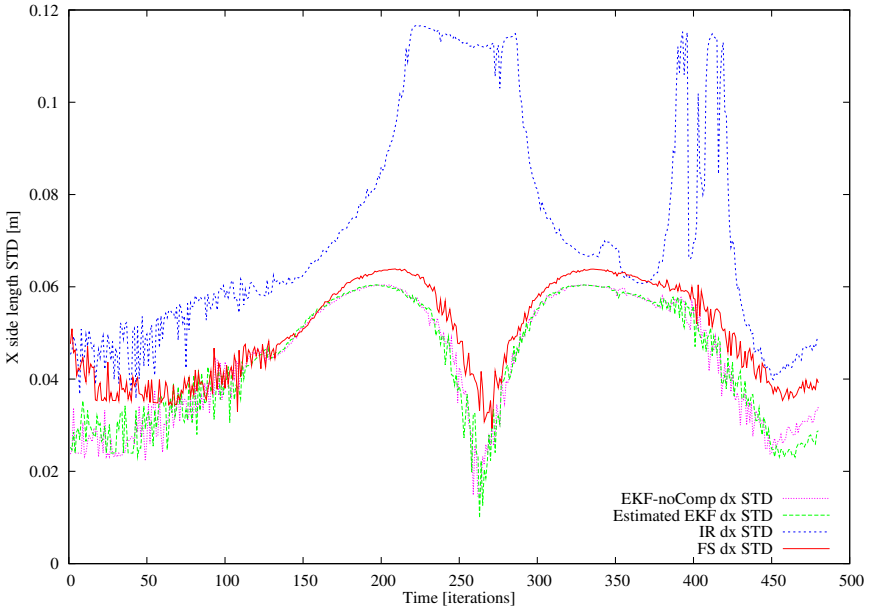

Figure 5.29: Size standard deviation (X coordinate).

respectively are shown. Figures 5.38 and 5.41 show the difference between the estimated velocity and the real one in X and Y coordinate. One can see in the aforementioned figures that the tracking without size change compensations produces bad velocity estimations, especially in places where size changes for a great deal in very short time.

As in the case of angular velocity estimation, the velocity estimate "follows" the real value, but does not converge with it. The explanation is

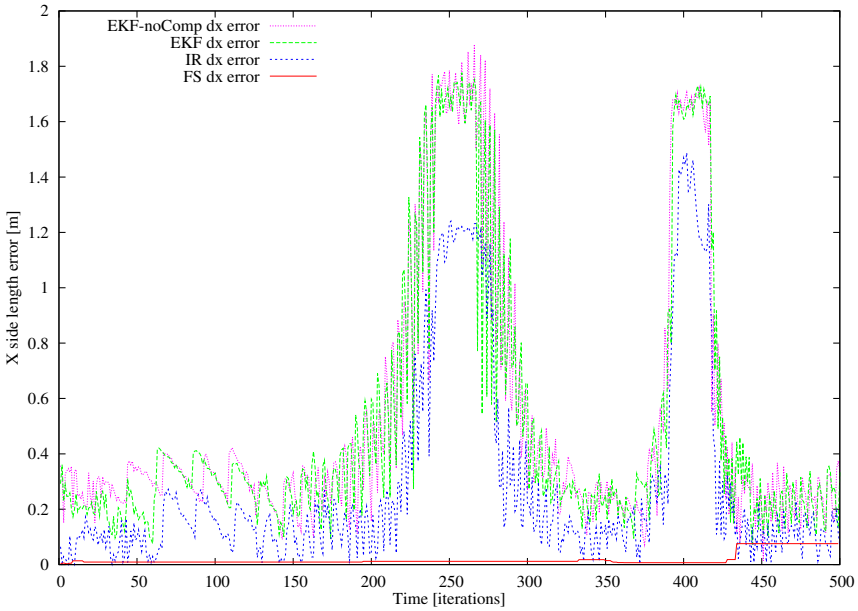

Figure 5.30: Difference between real and estimated sizes (X coordinate).

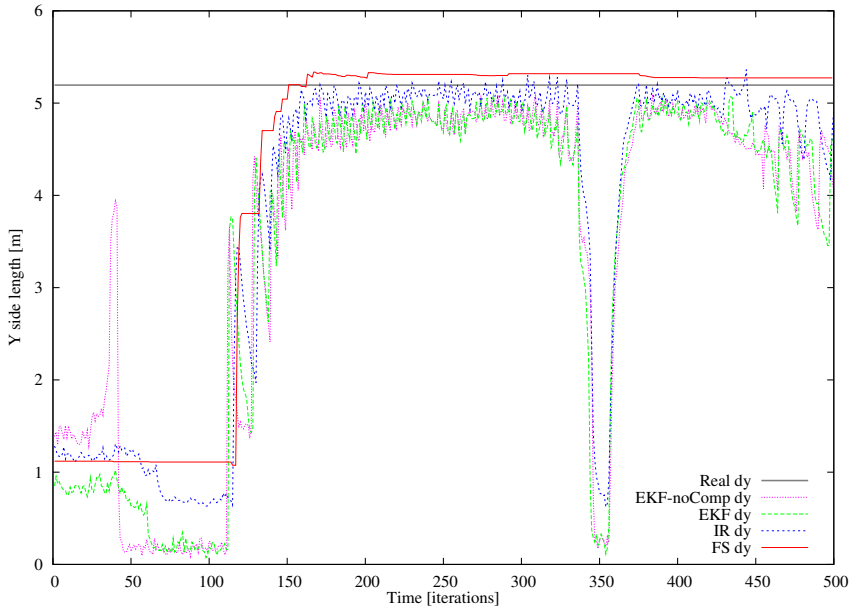

Figure 5.31: Size estimation (Y coordinate).

identical as in the angular velocity estimation. In the DWNA kinematic model, the transition equation preserves the velocity estimated in the previous filtering iteration. In the mentioned figure, one can see that the real angular velocity is changing all the time, and thus, the estimated value cannot converge to the real value.

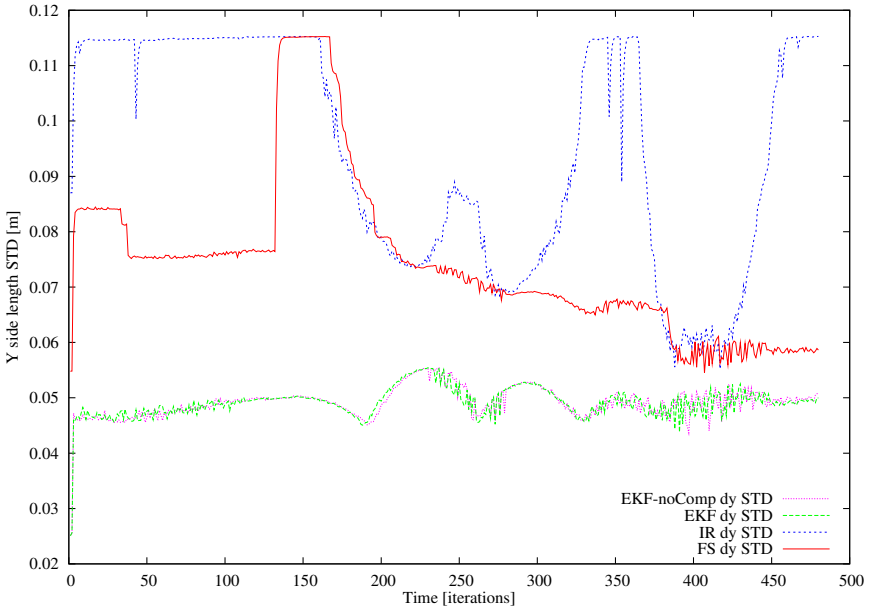

Figure 5.32: Size standard deviation (Y coordinate).

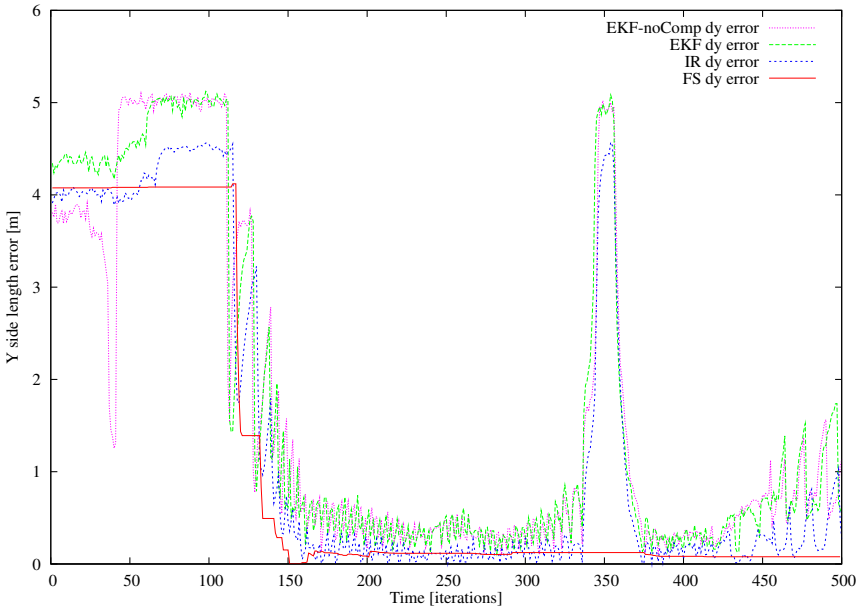

Figure 5.33: Difference between real and estimated sizes (Y coordinate).

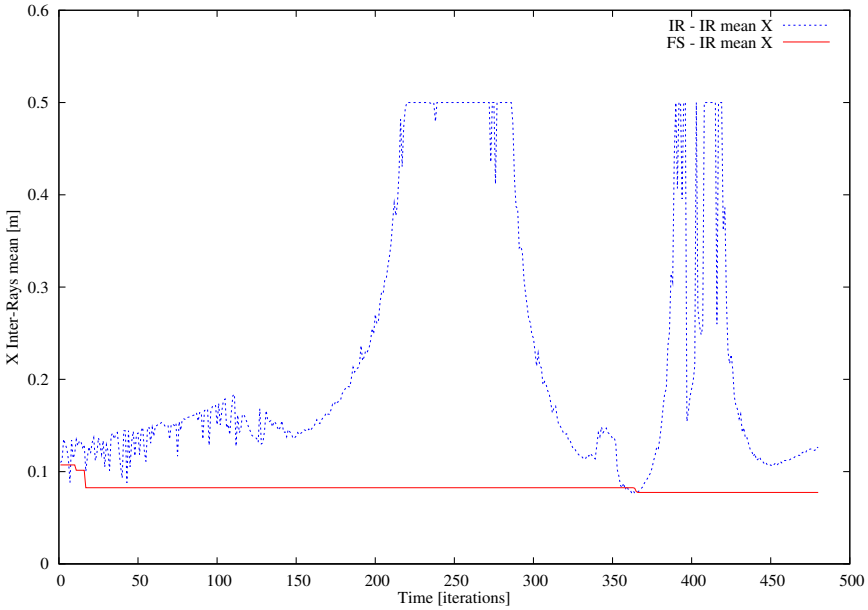

Figure 5.34: Inter-Ray mean (X coordinate).

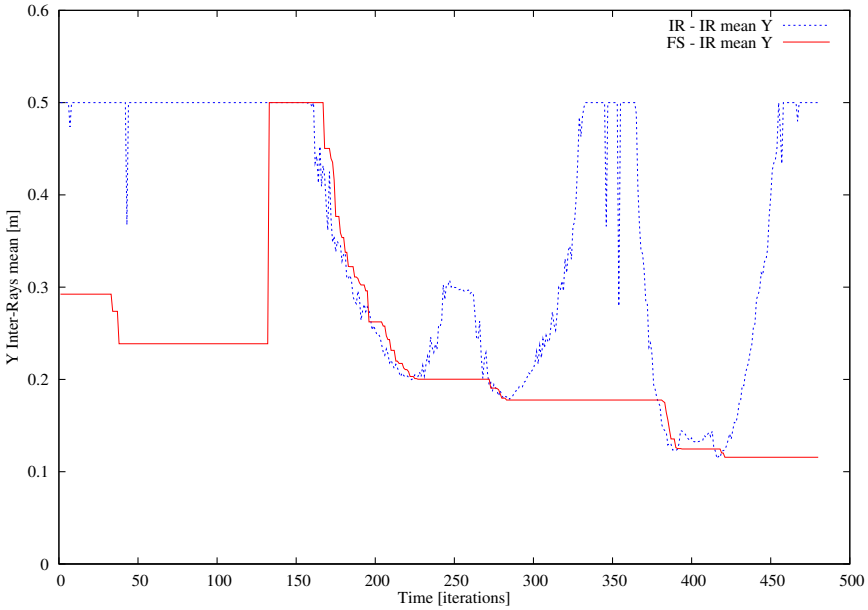

Figure 5.35: Inter-Ray mean (Y coordinate).

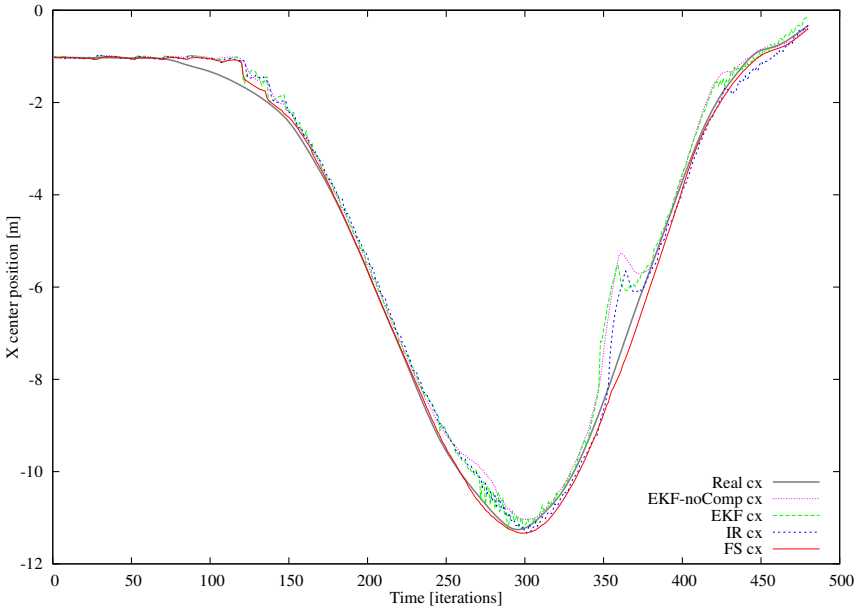

Figure 5.36: Center position estimation (X coordinate).

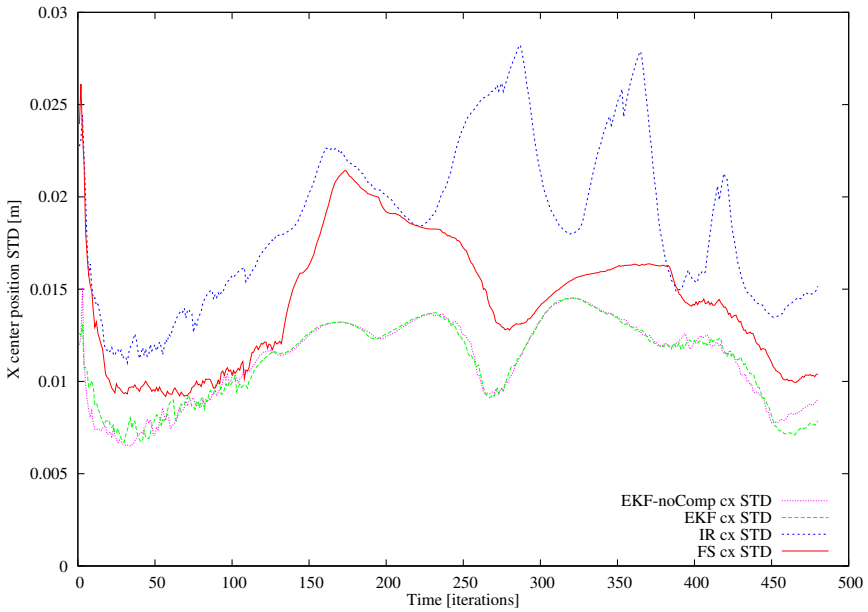

Figure 5.37: Center position standard deviation (X coordinate).

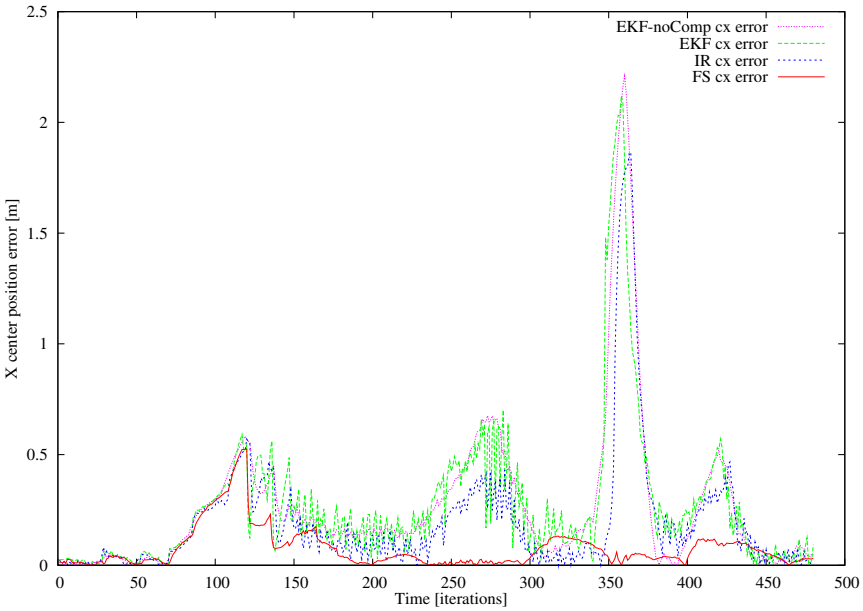

Figure 5.38: Difference between real and estimated Center positions (X coordinate).

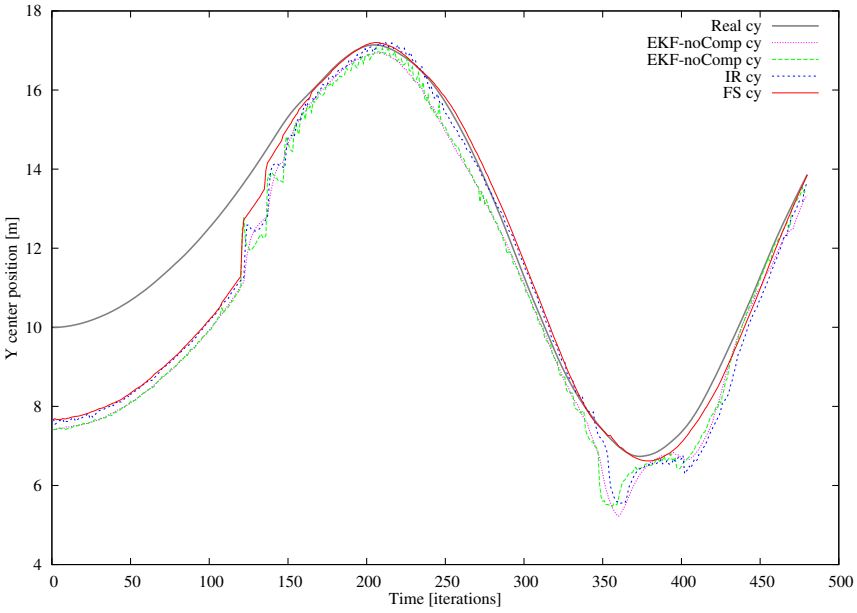

Figure 5.39: Center position estimation (Y coordinate).

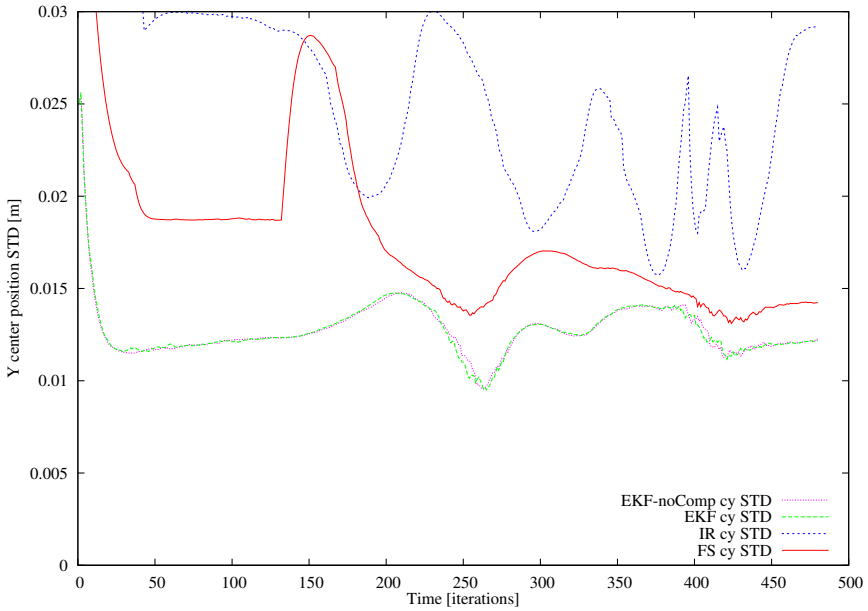

Figure 5.40: Center position standard deviation (Y coordinate).

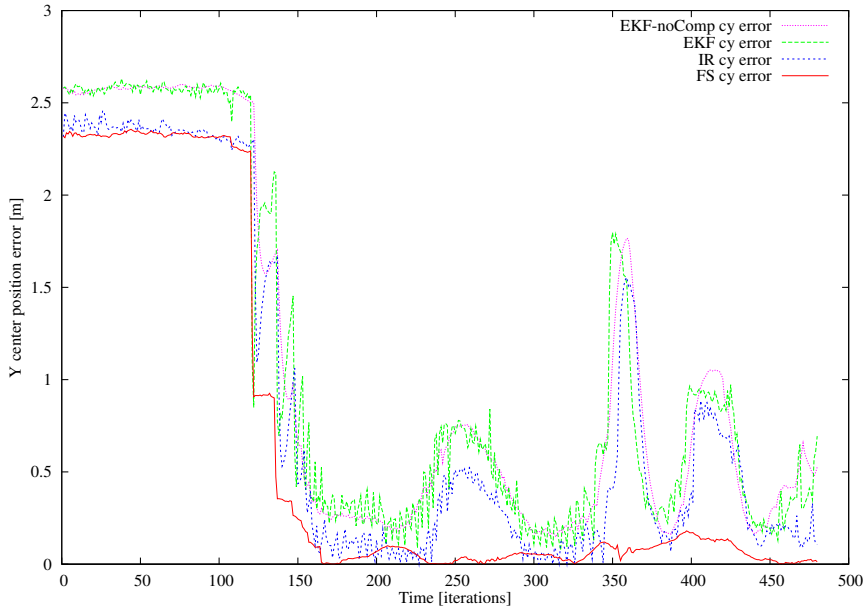

Figure 5.41: Difference between real and estimated center positions (Y coordinate).

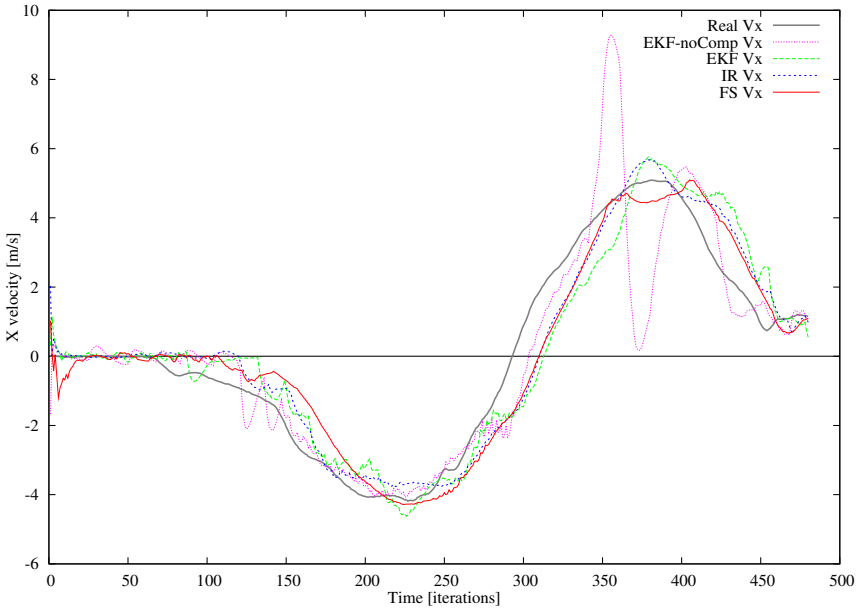

Figure 5.42: Linear velocity estimation (X coordinate).

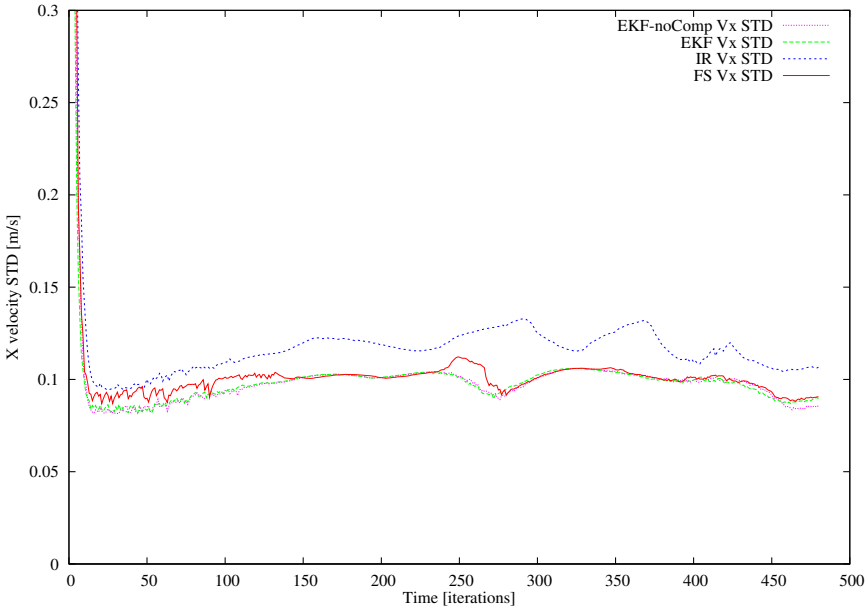

Figure 5.43: Linear velocity standard deviation (X coordinate).

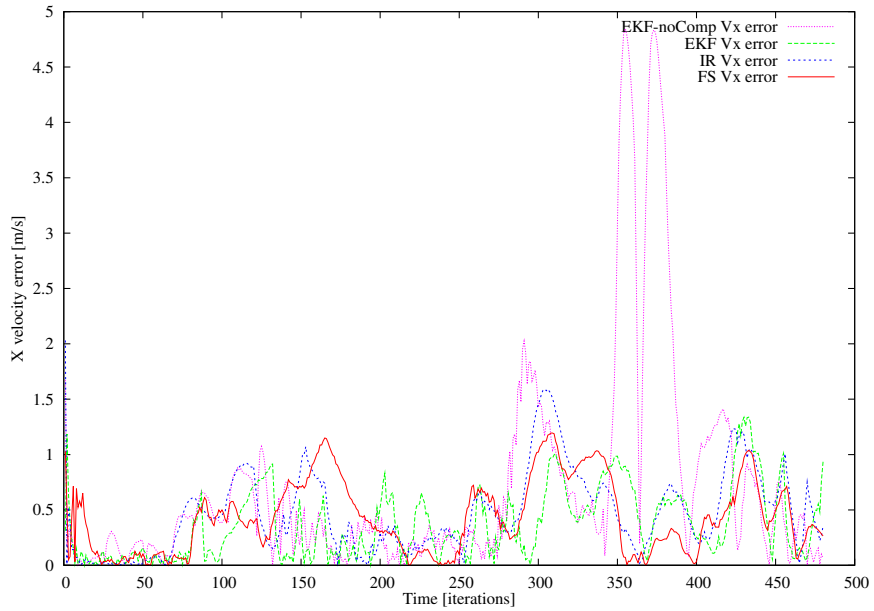

Figure 5.44: Difference between real and estimated linear velocities (X coordinate).

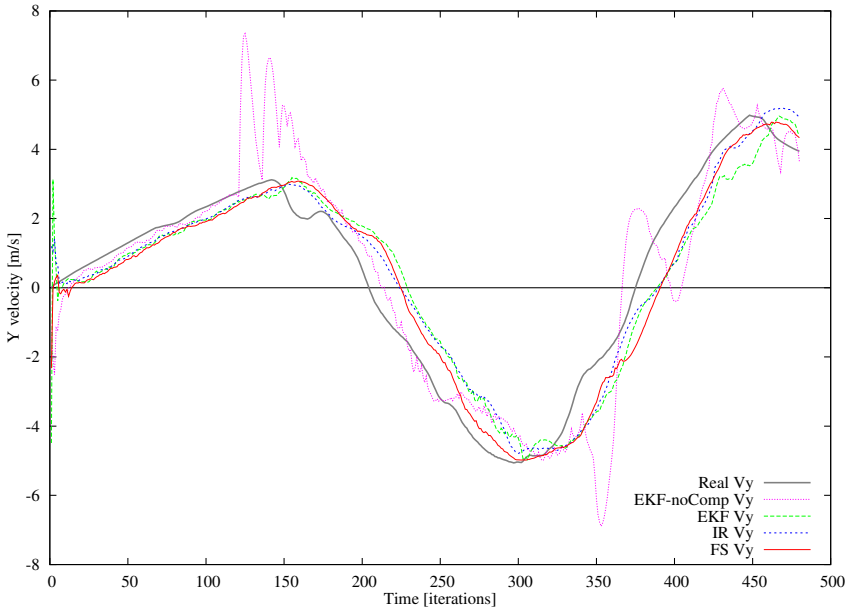

Figure 5.45: Linear velocity estimation (Y coordinate).

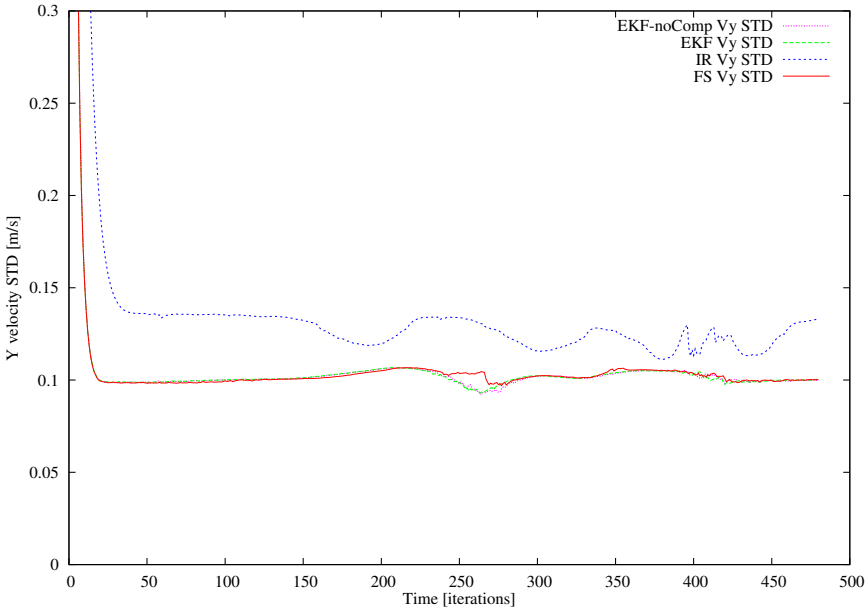

Figure 5.46: Linear velocity standard deviation (Y coordinate).

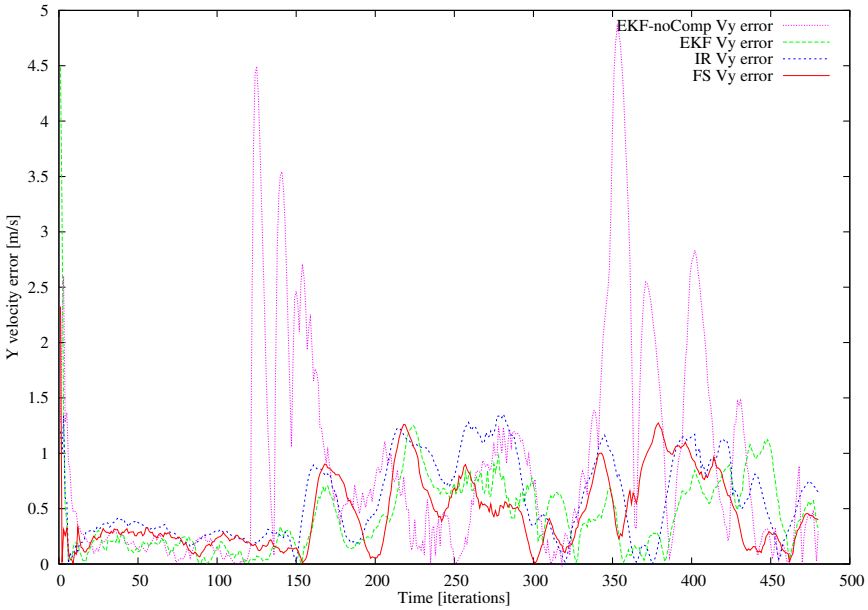

Figure 5.47: Difference between real and estimated linear velocities (Y coordinate).

5.5.2 Real vehicle platform based results

We evaluate the FS algorithm on real data. The data were acquired using the experimental vehicle. In the experiment, we use the same scenario as in the case of simulator based evaluation. The tracked vehicle is running in front of the instrumented vehicle following circle trajectory. The instrumented vehicle is equipped with laser scanner SICK LMS 221, which it's step angle is set to 1° with an angle range of 180°. To obtain the reference relative position of the tracked vehicle the three GPS-RTK configuration was used (see section 4.2).

The X axis in the figures represents the number of iterations of the tracking algorithm. The frequency of the sensor is set to 4 Hz, thus one iteration takes 0.25 seconds. The frequency of the sensor is low comparing to the real capability of the used sensor. This is due to the technical configuration at the moment of the running the tests. The experimental Intelligent Vehicle platform is still in evolution.

All the results presented in the graphs are expressed in the instrumented vehicle local coordinate system.

The lower frequency of the acquisition influences the tracking results, especially the angular and linear velocities.

Figure 5.48 shows the estimated trajectories and the real one of the tracked vehicle in the tested scenario. Like in the simulated data case, the FS algorithm outperforms the others in terms of position estimation precision. There are, however, places where the estimated trajectory deviates from the real one (in the extreme x locations)

The results of angle orientation estimation are presented in figures 5.49-5.51. Again, only the measurement, estimation of the angle and the real value are shown. In Figure 5.49, one can see that, between the real and estimated orientation angle, there is almost a constant gap. This is due to the effect of how the orientation of a real object is calculated. The real orientation angle was calculated by finding the slope of the line passing by the current and previous vehicle positions. This slop will never be collinear with real vehicle orientation when the vehicle is turning. Nevertheless, the gap between the two discussed values allows to conclude that the orientation angle is well estimated.

The IR and FS algorithms influence directly the estimation of the tracked object size. The size estimation results are shown in figures 5.52-5.59. Figure 5.52 and 5.55 present the size estimation of the tracked object. The FS produces a stable estimate. In the simulated data case the size was slightly

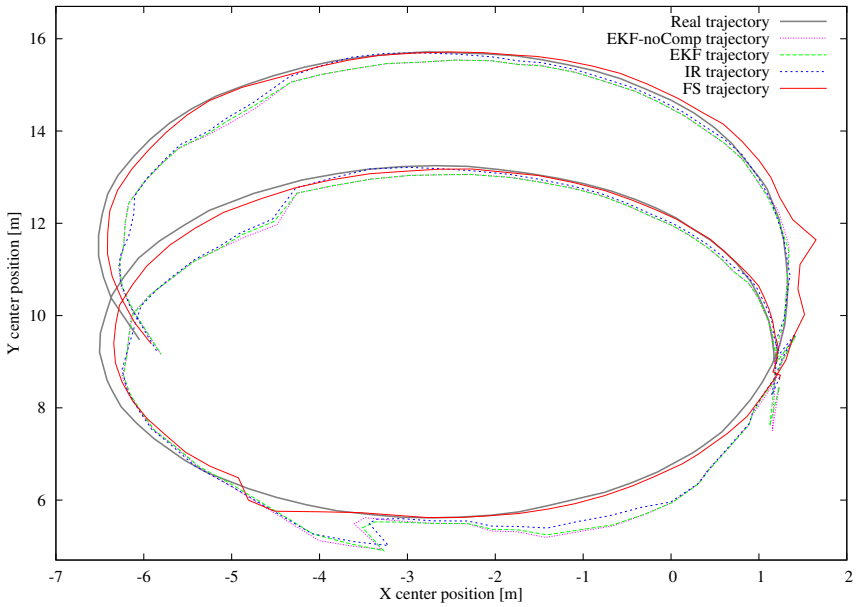

Figure 5.48: Trajectories of the tracked vehicle.

overestimated. In the presented case the values are underestimated. This is due to the fact that in the simulator, a special model of the vehicle has been prepared. It was prepared in a such manner that the vehicle profile obtained by cutting the model by a laser plane has the maximal vehicle observable size. In the real experimentation, however, the estimated value was compared with information provided by the manufacturer. In other worlds, we cannot be sure that the profile obtained by cutting by a laser plane has the maximal size of the vehicle. The estimation error is presented in figures 5.54 and 5.57. The FS algorithm provides most precise estimation. The places when the estimation deviates from the real object size, correspond to the situations when only one side of the object can be perceived by the sensor. Even in this situations, while using the FS, the best size estimate is available.

Figures 5.53 and 5.56 present the size estimation standard deviation. The same characteristics, as in the case of the simulated data, can be observed. The usage of the IR uncertainty results in greater STD values for the IR and FS algorithms. The FS based size STD decreases along the decrease of the stored IR value.

The results of position estimation can be seen in figures 5.60 - 5.62. Figures 5.60 and 5.63 show the estimated and real object positions in X and Y coordinate. Figures 5.62 and 5.65 show the difference between the

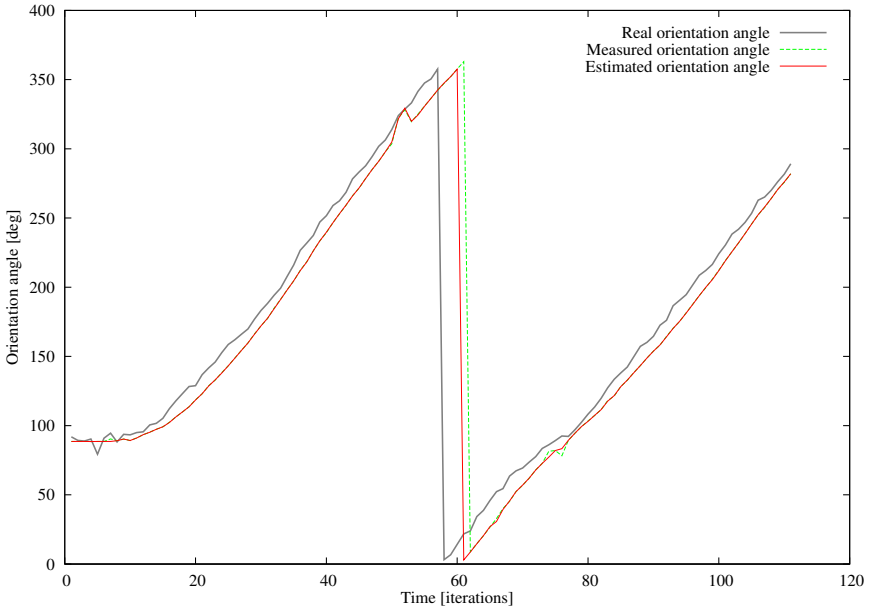

Figure 5.49: Orientation angle estimation.

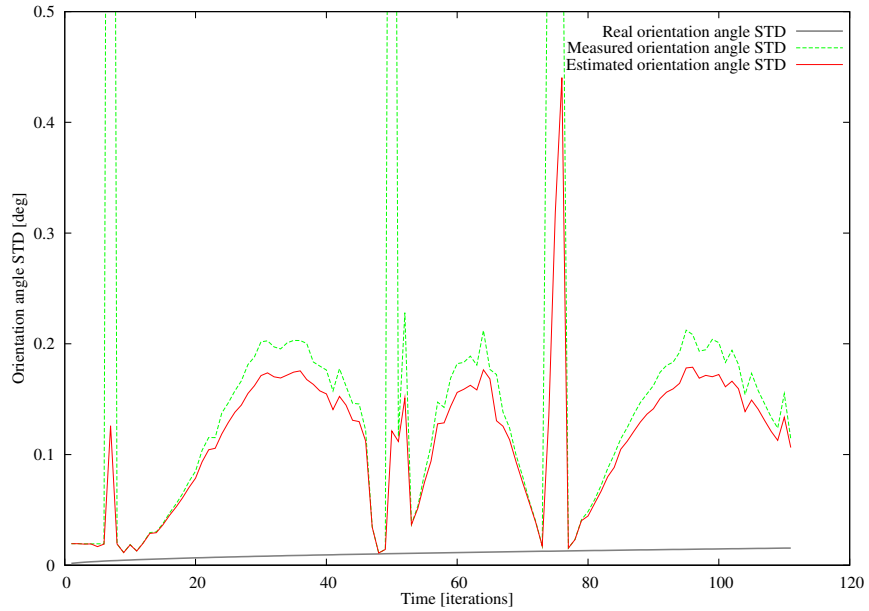

Figure 5.50: Orientation angle standard deviation.

estimated position and the real one in X and Y coordinate. The obtained results, presented in mentioned figures confirm the results obtained with the simulated data. The FS approach provides the best center estimation. Figures 5.61 and 5.64 show the standard deviation (STD) of the position estimation in X and Y coordinate. Since the position uncertainty is related

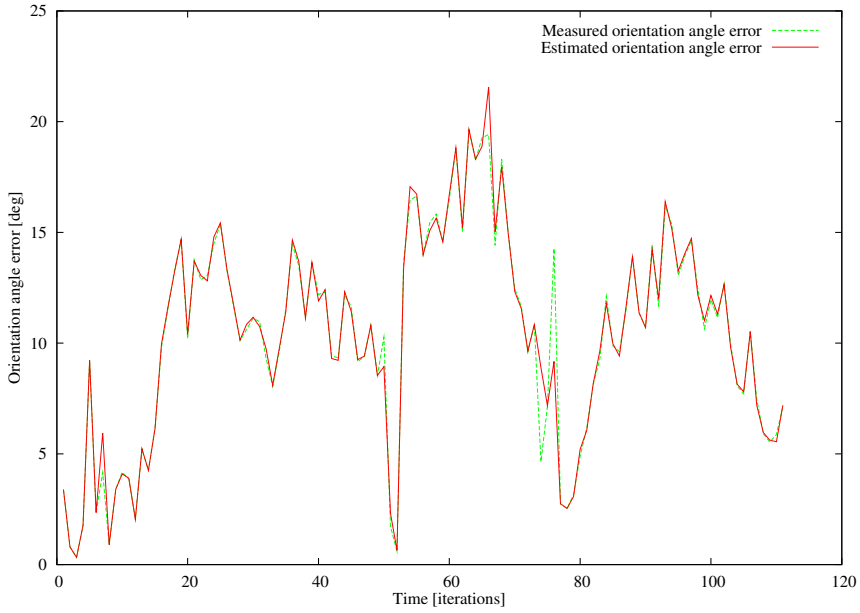

Figure 5.51: Difference between real and estimated orientation angles.

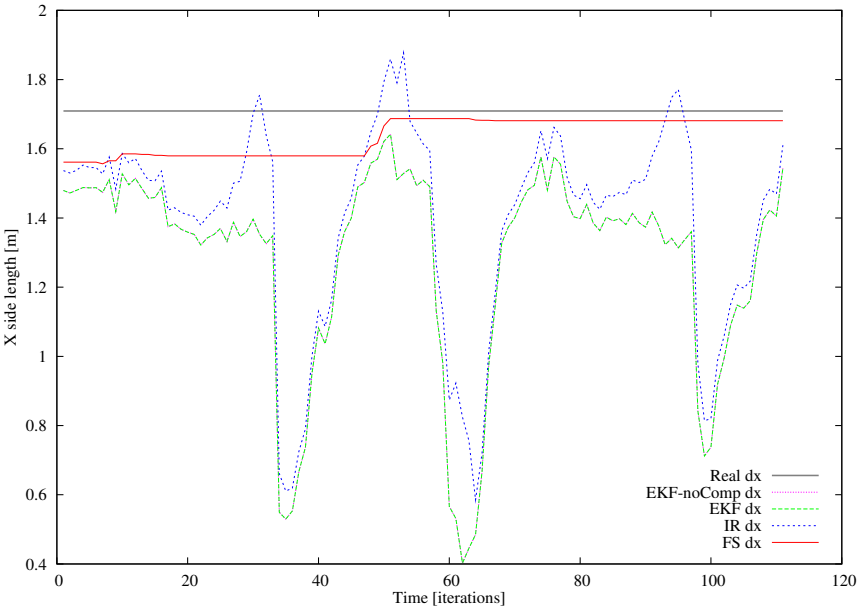

Figure 5.52: Size estimation (X coordinate).

with size the STD, the introduction of the IR uncertainty effects in the increase of position standard deviation in the IR and the FS algorithms. In the case of the FS algorithm, the size STD decreases in time (it is well observable in Figure 5.64). This is due to the decrease of the IR value in the corrected measurement.

Figures 5.66-5.71 show the velocity estimation results. In figures 5.66 and 5.69 the estimated and real object velocities in X and Y coordinate

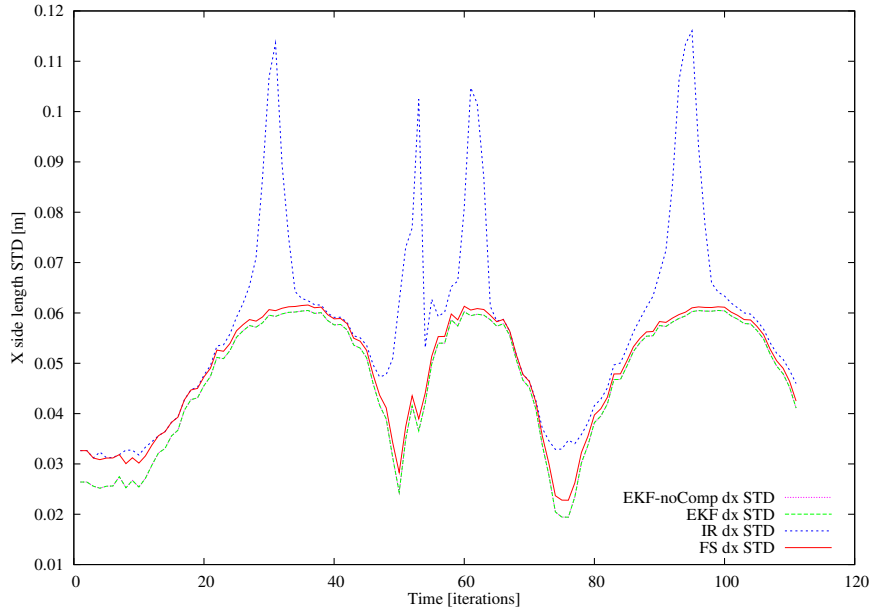

Figure 5.53: Size standard deviation (X coordinate).

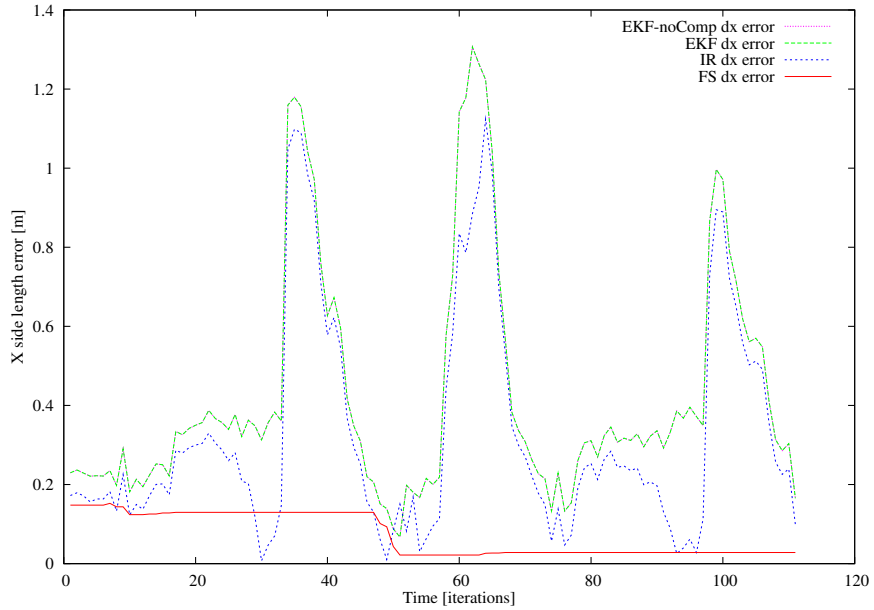

Figure 5.54: Difference between real and estimated sizes (X coordinate).

are shown. Figures 5.62 and 5.65 show the difference between the estimated velocity and the real one in X and Y coordinate. One can see, in the aforementioned figures that effect of "phantom" movements on velocity estimation is visible in the case of "EKF-noConp" algorithm (the approach in which no size change compensation was applied). Compared to the simulated data, there is no gap between the real and estimated velocities. This is an effect of low frequency of observations. Each *a priori* covariance of

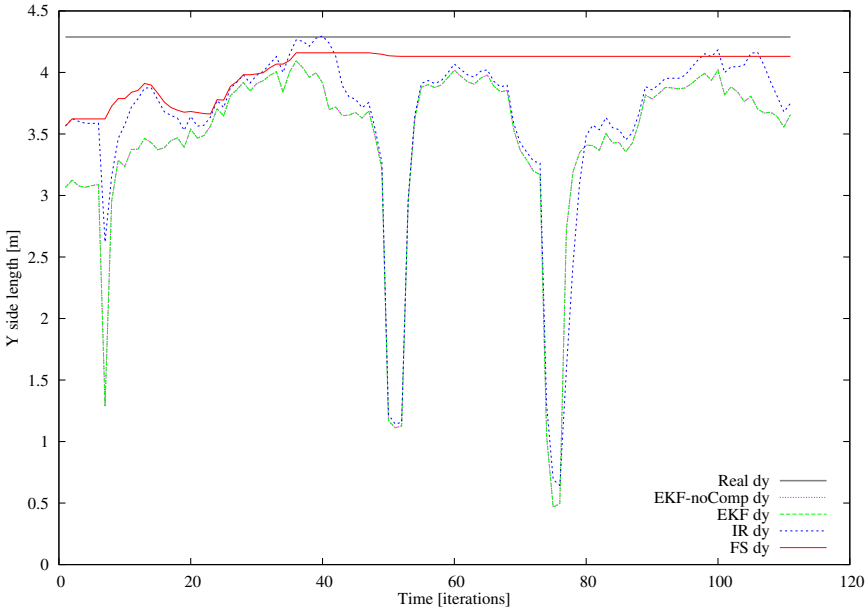

Figure 5.55: Size estimation (Y coordinate).

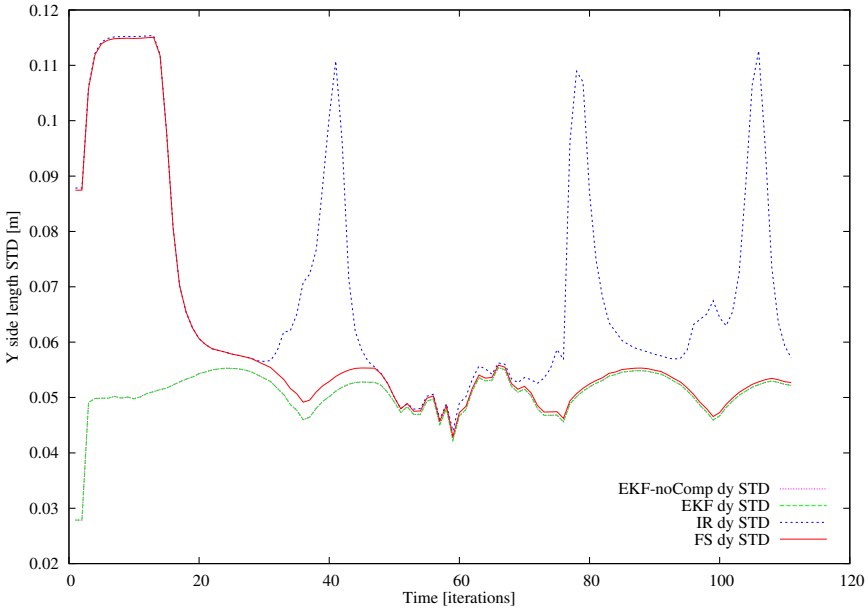

Figure 5.56: Size standard deviation (Y coordinate).

the velocity is more noisy in this case than in the simulated data case. The *a priori* velocity variance is proportional to the process noise (modeled by the acceleration noise in used the DWNA) and the time passed during the measurements. In the simulated case, the sensor provides data 50 time per second, what gives time delta equal to 0.02 seconds. The real sensor has around 4Hz, what gives time delta equal to 0.25 seconds. The same process noise values were used in the two cases. Thus, the *a priori* velocity variance

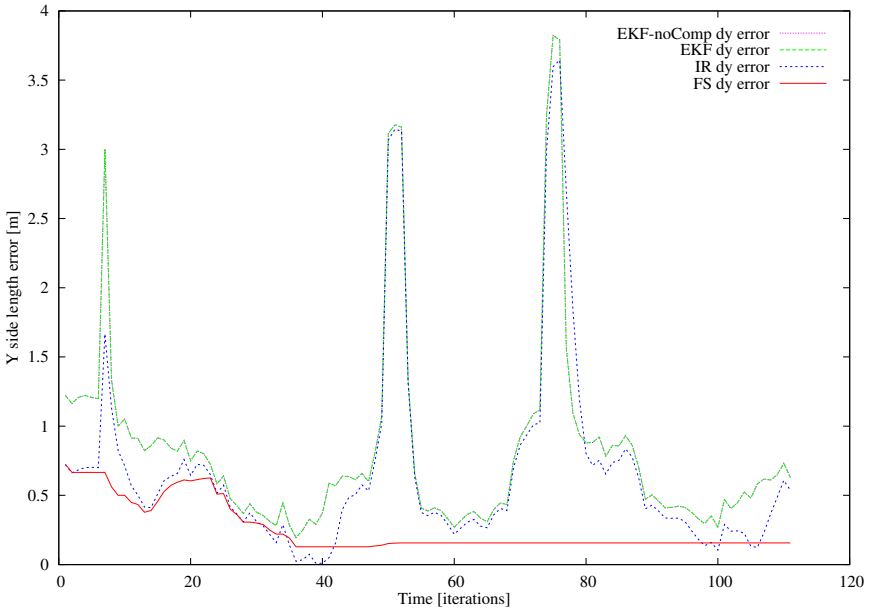

Figure 5.57: Difference between real and estimated sizes (Y coordinate).

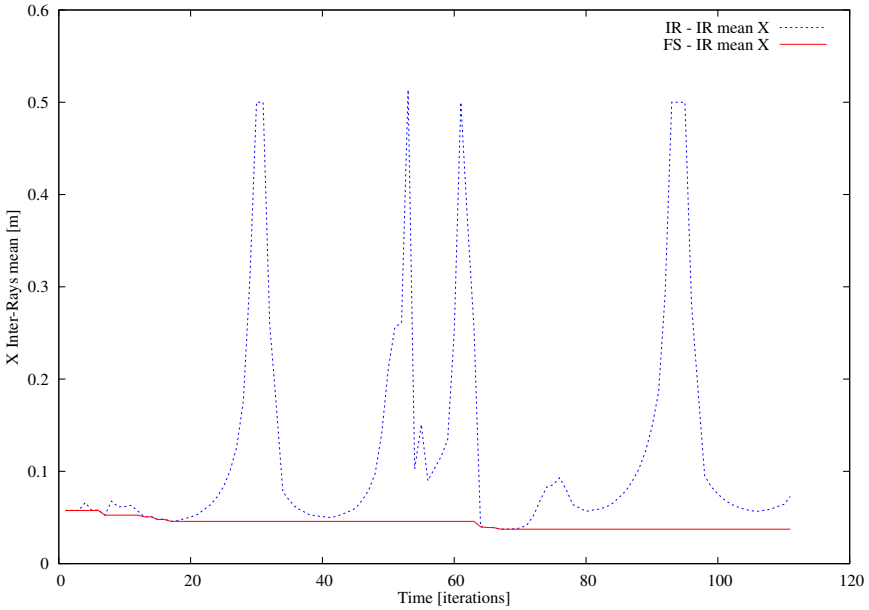

Figure 5.58: Inter-Ray mean (X coordinate).

in the real data case is much greater than in simulated case. The greater variance of the predicted velocity causes, that in the Kalman Filter, the measurement has more influence on corrected estimate than the predicted velocity.

To illustrate this effect, we simulated the tracking process with the following frequencies 50Hz and 4Hz, for the same scenario. Figures , and show the velocity estimation for data frequency 50Hz. Figures and show velocity

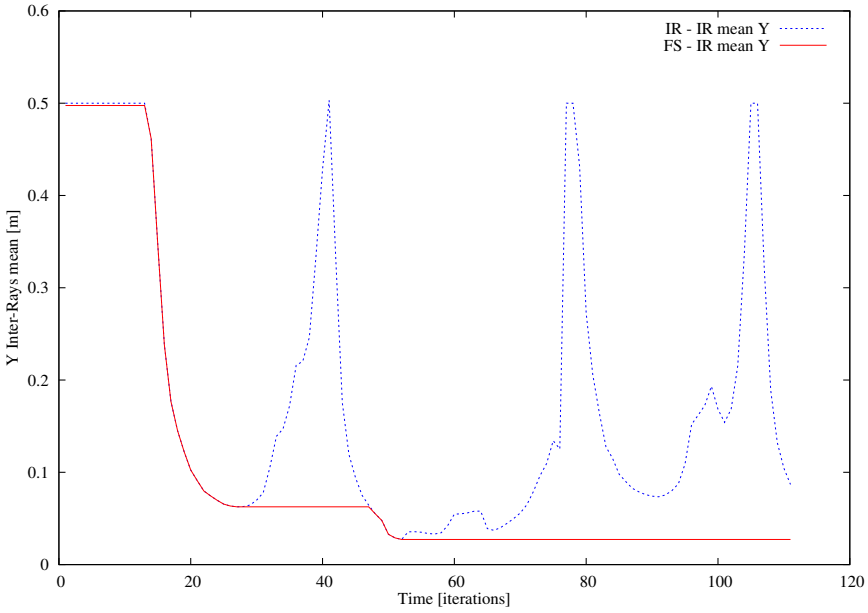

Figure 5.59: Inter-Ray mean (Y coordinate).

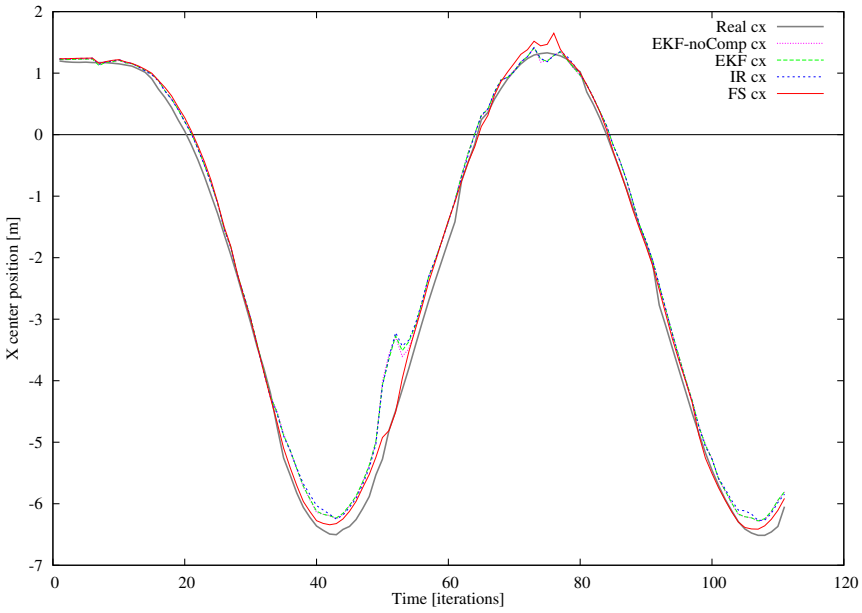

Figure 5.60: Center position estimation (X coordinate).

estimation for frequency 4Hz. One can see that the results of the velocity estimation with 4Hz are similar to the one obtained in the real data case. One may have an impression that the lower frequency provides the better velocity estimation, but this is not the case. To obtain better velocity estimation for data with higher frequencies, the process noise should be changed

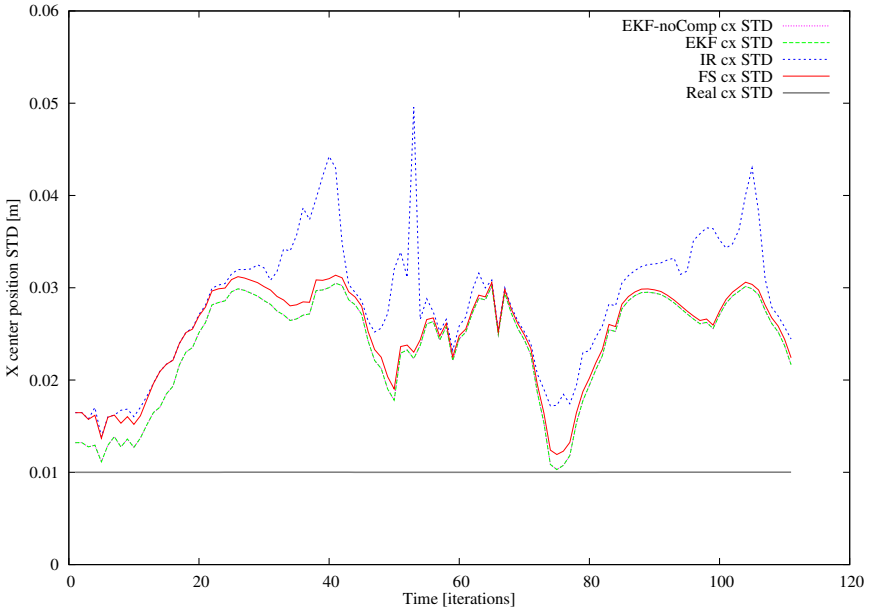

Figure 5.61: Center position standard deviation (X coordinate).

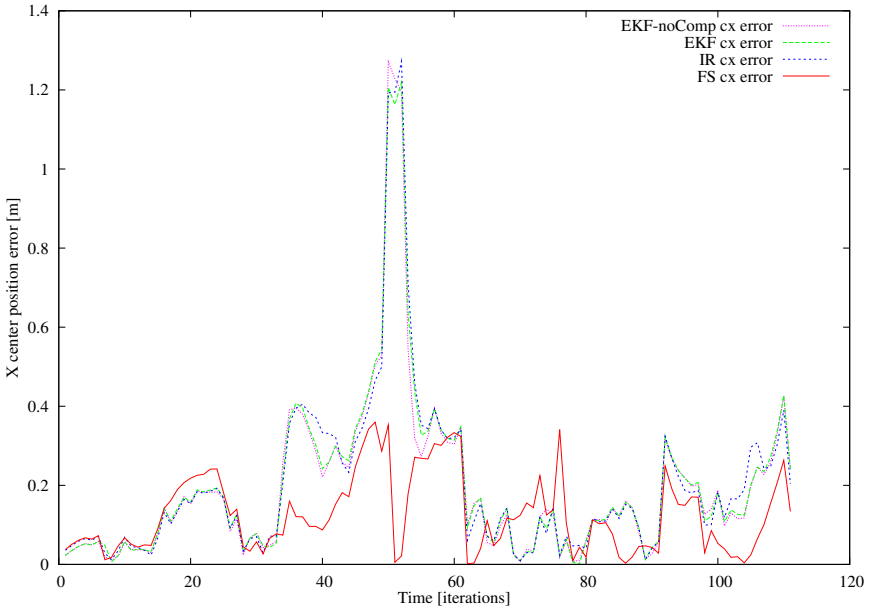

Figure 5.62: Difference between real and estimated Center positions (X coordinate).

and adapted to the used frequency. Another solution would be the usage of other kinematics models, which estimate also acceleration. The Discrete Wienner Process Acceleration model can be used.

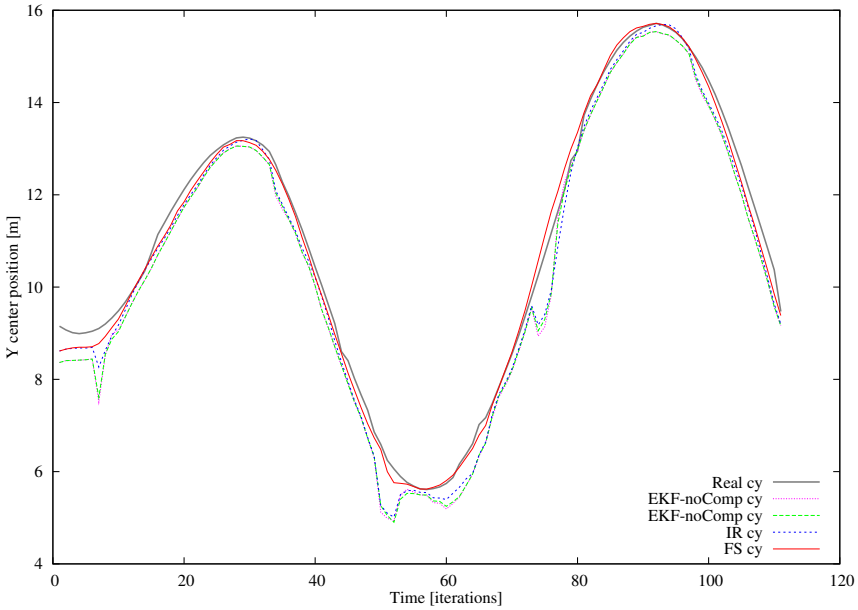

Figure 5.63: Center position estimation (Y coordinate).

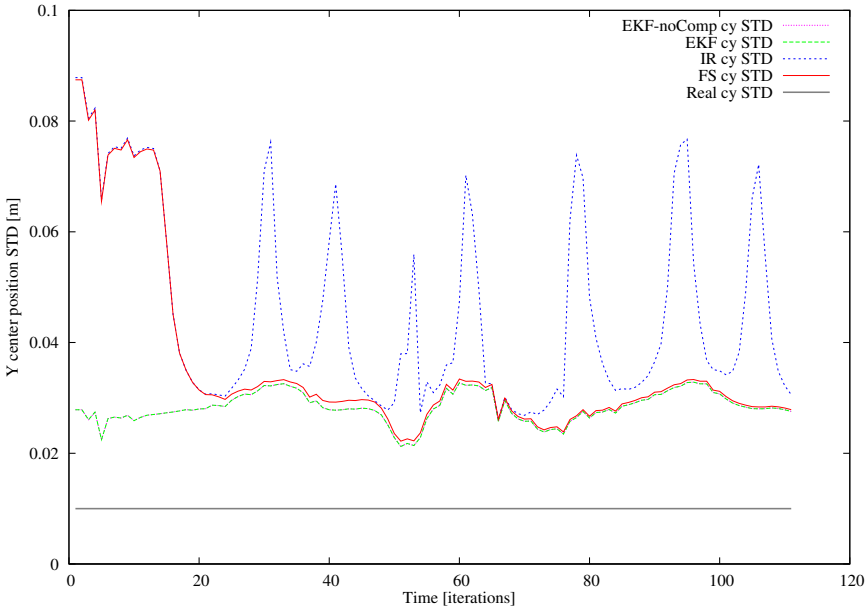

Figure 5.64: Center position standard deviation (Y coordinate).

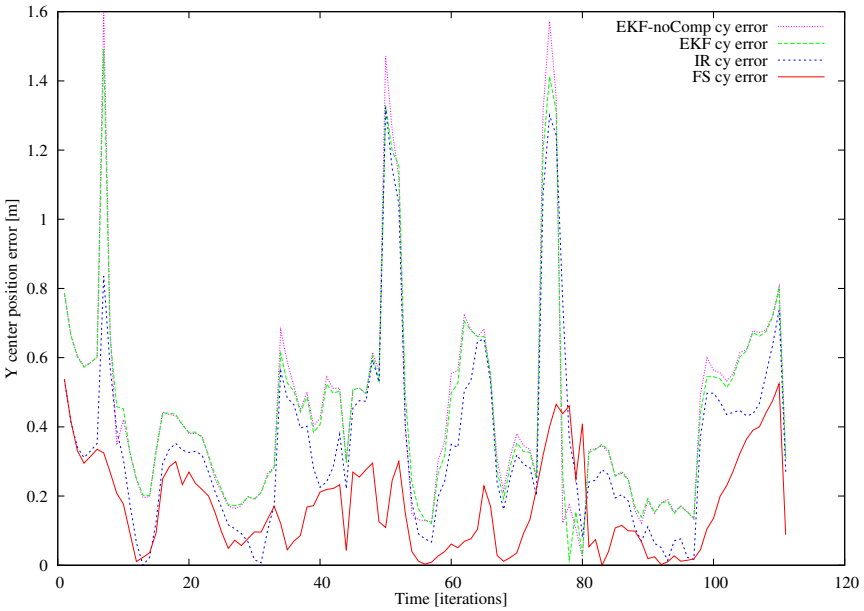

Figure 5.65: Difference between real and estimated center positions (Y coordinate).

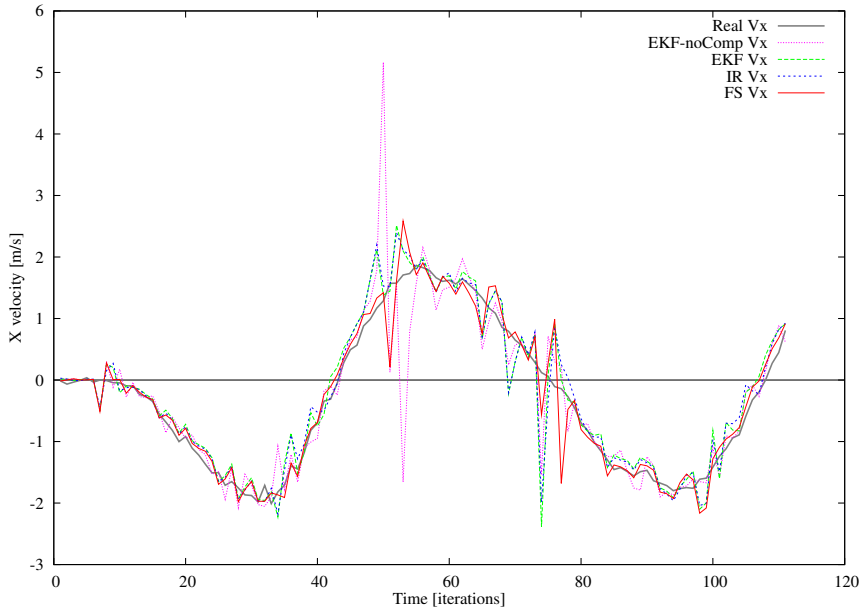

Figure 5.66: Linear velocity estimation (X coordinate).

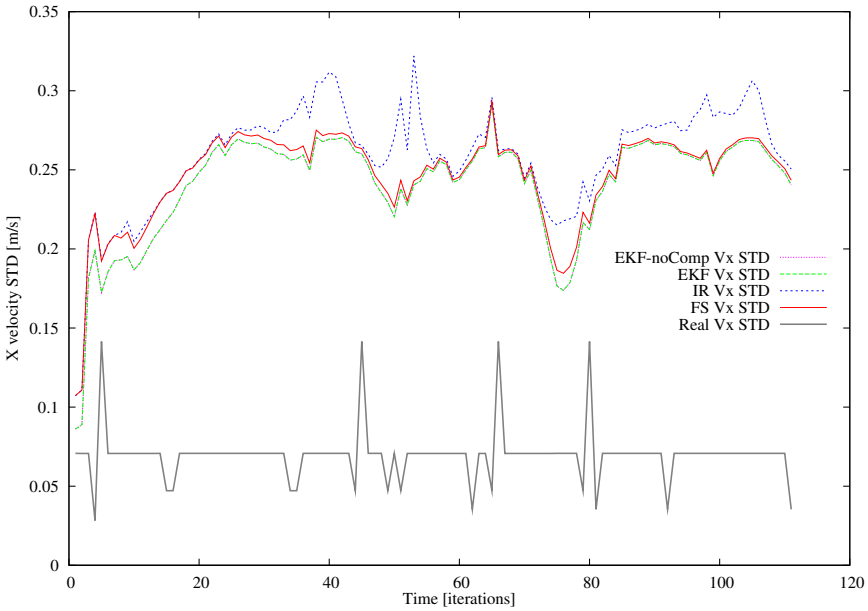

Figure 5.67: Linear velocity standard deviation (X coordinate).

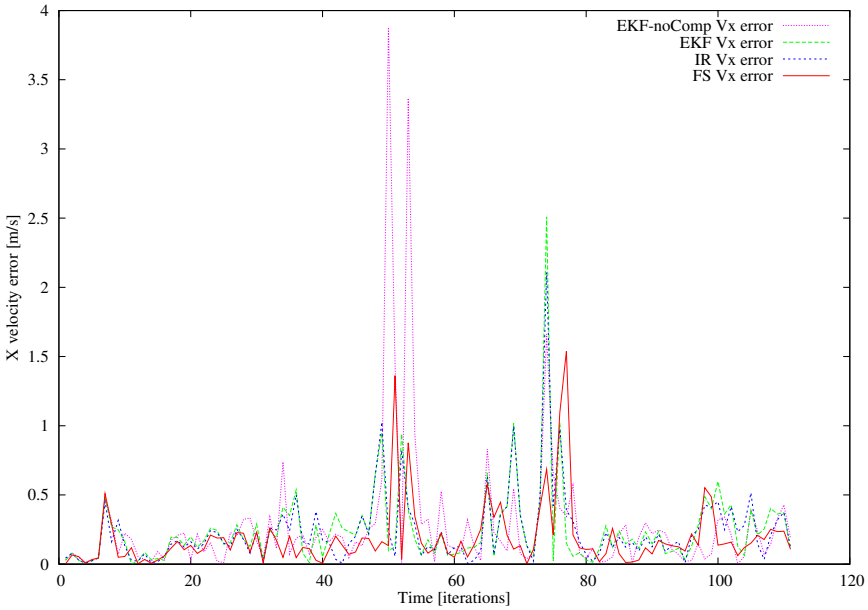

Figure 5.68: Difference between real and estimated linear velocity (X coordinate).

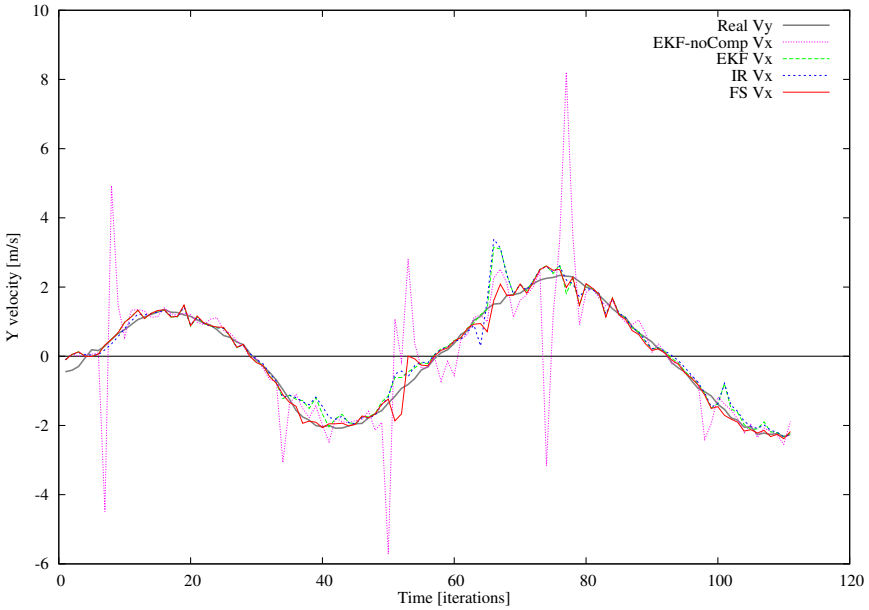

Figure 5.69: Linear velocity estimation (Y coordinate).

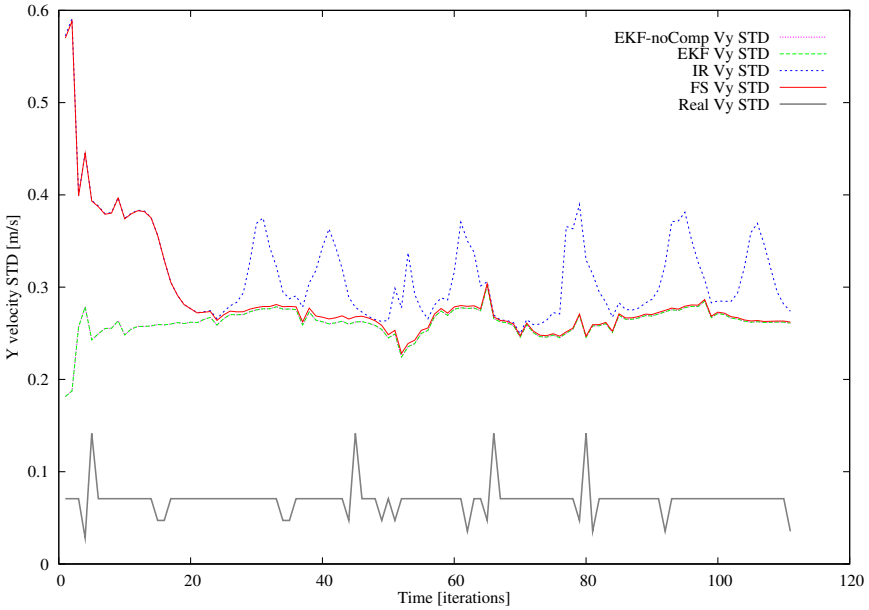

Figure 5.70: Linear velocity standard deviation (Y coordinate).

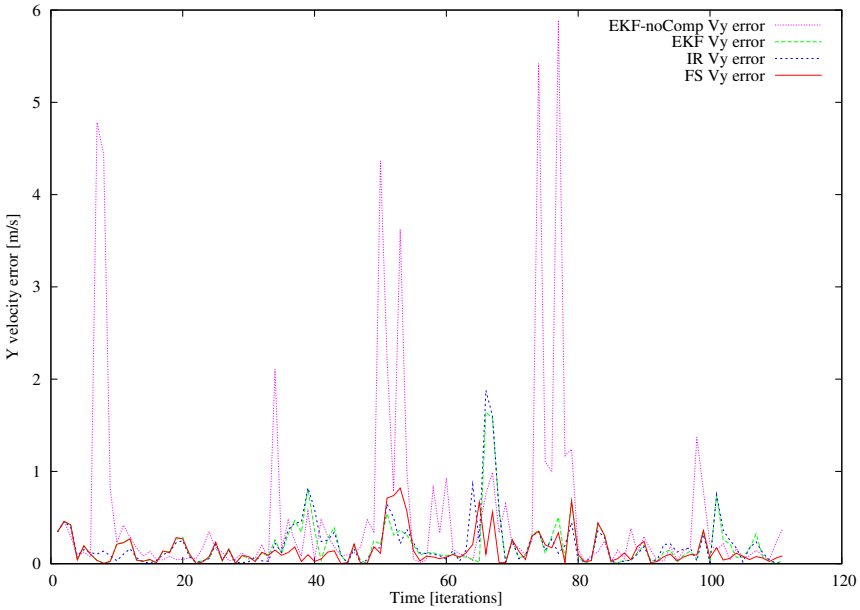

Figure 5.71: Difference between real and estimated linear velocity (Y coordinate).

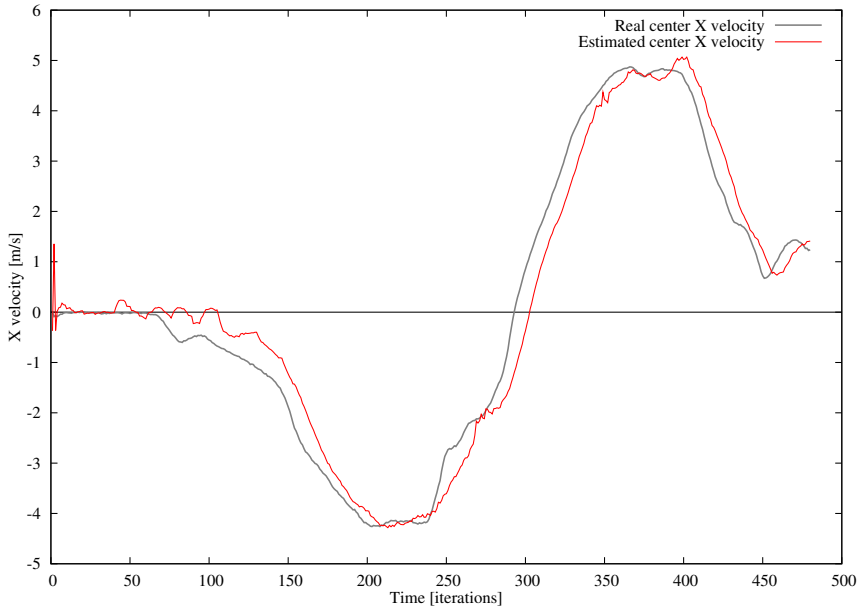

Figure 5.72: Linear velocity estimation for simulated data obtained with frequency 50Hz (X coordinate).

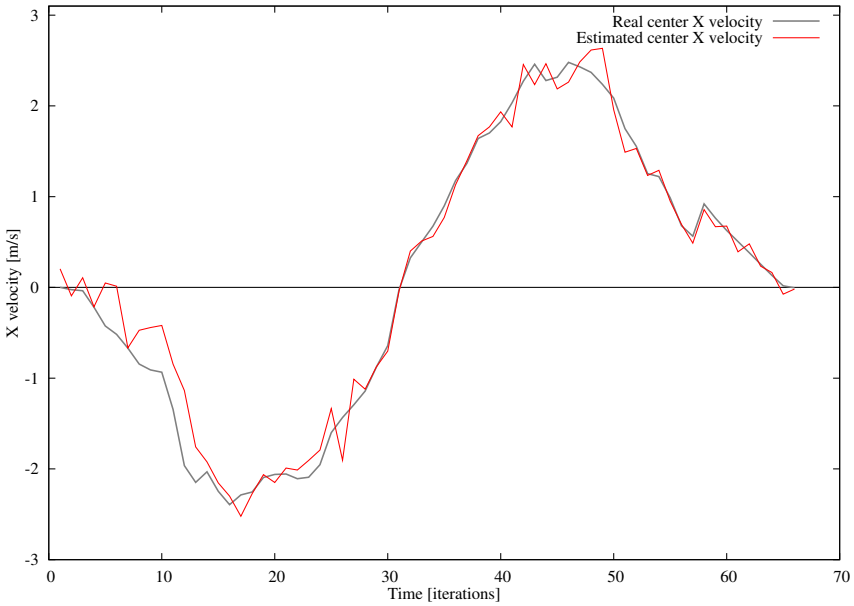

Figure 5.73: Linear velocity estimation for simulated data obtained with frequency 4Hz (X coordinate).

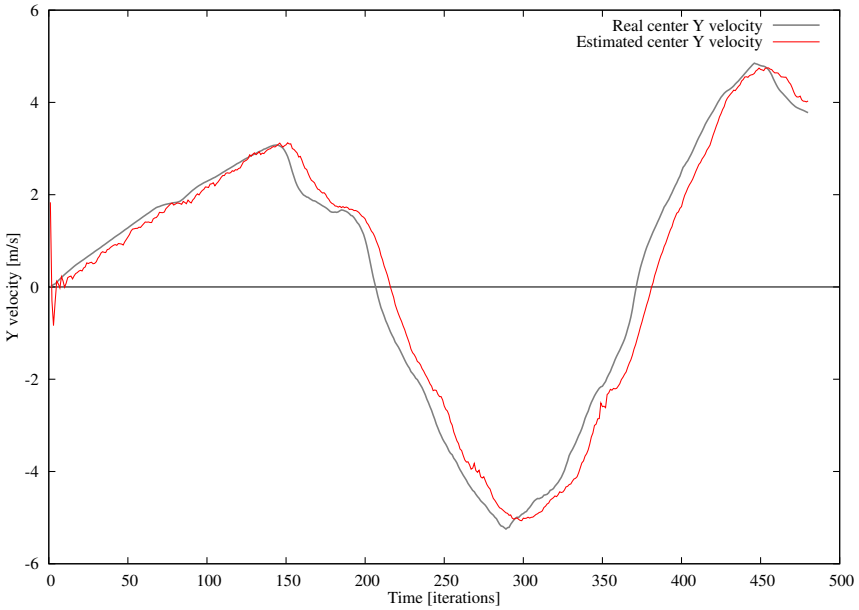

Figure 5.74: Linear velocity estimation for simulated data obtained with frequency 50Hz (Y coordinate).

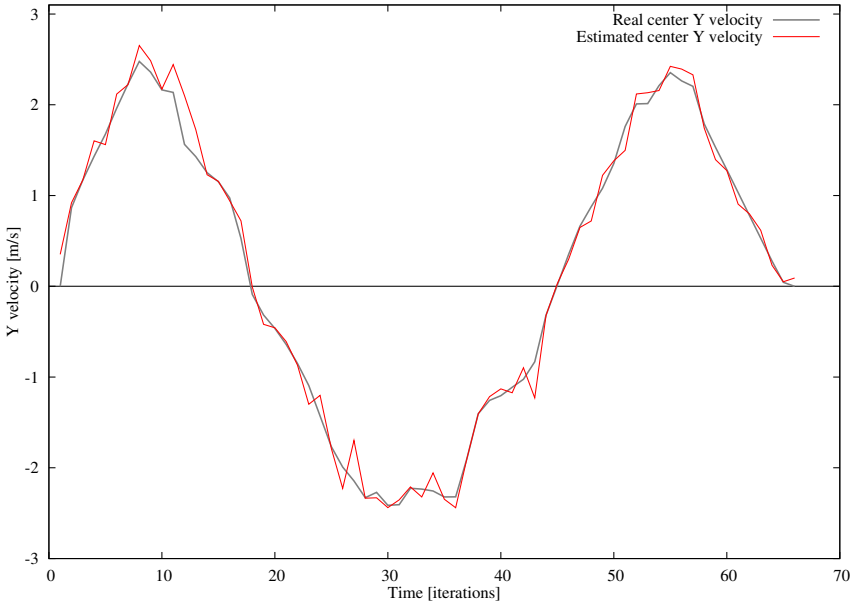

Figure 5.75: Linear velocity estimation for simulated data obtained with frequency 4Hz (Y coordinate).

5.5.3 Conclusions

In the last four sections, the methodology for object representation was introduced. The results, comparing proposed algorithms, was presented. It was shown that the proposed OBB based object representation method is suitable for the task of dynamic objects tracking. To increase the reliability of the tracking, two algorithms were introduced Inter-Rays based and Fixed Size based algorithm. By introducing the size uncertainty, the first algorithm allows to represent better the real size and position uncertainties. The usage of the Fixed Size assumption allows to obtain better tracking in terms of size and center position estimation. The results obtained using the simulator was confirmed by the processing of real data. The better size estimate can be exploited in data association. Combination of a quick method with the FS algorithm, which results in robust method, is presented in the next chapter.

5.6 Two laser scanner fusion

In this section, we propose a two laser scanner based fusion approach that exploit better the angular resolution of the two laser scanners configuration. The objective is to increase the tracking accuracy in terms of object size, angle orientation, velocity estimation.

We have shown, in section 5.1, that representation model performs well except for objects poorly represented by raw data points. This situation occurs for far objects. Indeed, the number of laser rays colliding with objects is inversely proportional to the distance and proportional to the LRF angular resolution. Since the increase of the LRF angular resolution is limited, the number of laser rays colliding with objects decreases with the distance. Hence, at a certain range, the objects state estimation becomes very uncertain or even impossible to obtain. To overcome this limitation, more LRF sensors can be used. A multiple LRF configuration provides a higher perception angular resolution, and thus, a better object state estimation can be achieved. Furthermore, interlacing rays allows an additional size estimation refinement by utilizing Inter-Rays uncertainty.

KF based fusion methods can be divided into two groups: measurements fusion and tracks fusion. In the case of far objects, none of general approaches fits. In [46], two LRF were fused by using Weighted Measurement Fusion (WMF) method [30] is proposed. In this method, OBB measurements are extracted from raw data points for each sensor. The OBB measurements coming from the two sensors are extracted and then fused. This method takes into account only the redundancy aspect of the two-LRF configuration, and does not benefit from the increased perception angular resolution. Thus, it does not perform well for far objects. A method taking into account the redundant aspect of the multi-sensor configuration must operate on raw sensory data.

In this section, we propose a two LRF based fusion approach that takes advantage from the increased perception angular resolution (more raw data points per object and lower distance between laser rays). To benefit from this aspect, the raw data points coming from the sensors must be merged to extract an OBB measurement. The first step of the whole tracking system consists of data association. Raw data points association is performed for each sensor separately, and raw data points are regrouped in clusters. The number of clusters correlated with a track is equal to the number of sensors. During the points clustering, the online semi convex-hull construction takes place (see section 5.1.1). The points constructing each semi convex-hull are

sorted according to their angular coordinate. To construct the fused semi convex-hull from the semi convex-hulls correlated with a track, the following algorithm is performed (see Figure 5.76). It starts by inserting the two points with the smallest angular coordinates into a new semi convex-hull to be constructed. To choose a point with the smallest angular coordinate, we consider only the first points of all the semi convex-hulls, since the points of each semi convex-hull are sorted. The point being inserted is deleted from the original semi convex-hull. In each iteration, a new point with the minimum angular coordinate is inserted into the semi convex-hull being constructed. For each point insertion, the convexity condition is checked. If this condition is violated, the existing semi convex-hull recalculation occurs (see section 5.1.1). The constructed semi convex-hull serves then as an input for the Calipers based OBB extraction method.

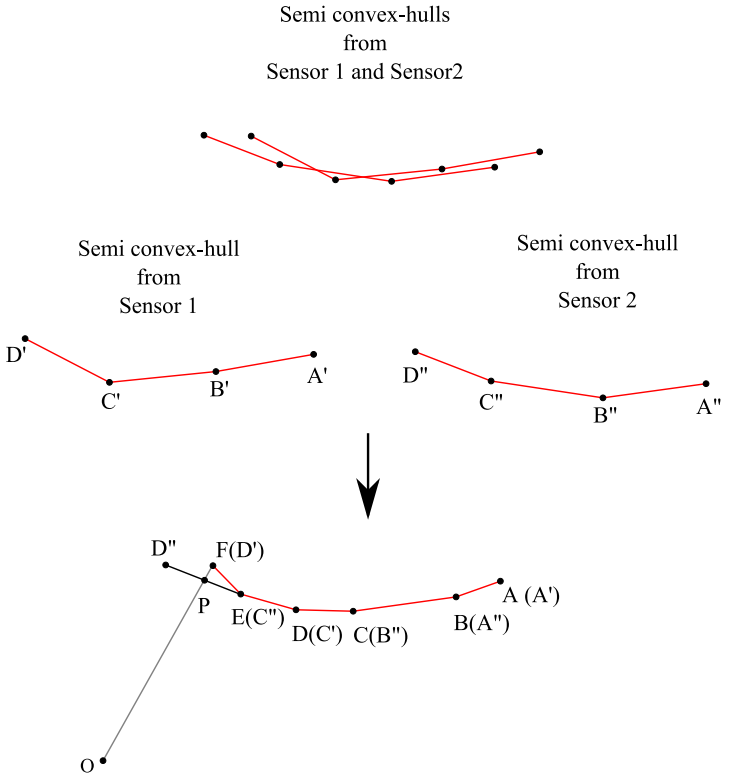

Figure 5.76: Semi convex hulls fusion.

After the OBB extraction, the Inter-Ray (IR) based size refinement starts. In the case of a single LRF, the distance between rays increases with the distance from the sensor. In a mulit-sensor case, the inter-rays distance varies between 0 and d_{LRF}, where d_{LRF} is a inter-rays distance

for each LRF. This IR distance variation allows to refine the size of the perceived objects, where the refinement level corresponds to the relative position between the objects and the sensors.

The IR uncertainty computation for multiple sensors is similar to the single sensor case (see section 5.3.2). The only difference between the two configurations is that in the multiple sensor case, the Inter-Rays distance d_{IR} values are computed for each LRF and than the smallest one is chosen. To correctly choose rays r and $r + n$ of each sensors, the coordinates of the extreme point (e.g. $maxY$) must be expressed in the local sensor coordination system (see Figure 5.17).

5.6.1 Simulator based results

To test the proposed approach, a single LRF and a two-LRF configurations are evaluated and compared. In the first configuration, a Laser Range Finder (LRF) is mounted in front of the instrumented vehicle. In the second configurations, two LRFs are parallely mounted in front of the vehicle with a horizontal inter-space of 1m. The step angle for the LRFs is set to 1° with an angle range of 180° (similarly to the real sensor parameters). The sensor range is set to 120 m and the range uncertainty σ_ρ is set to 0.05m.

To evaluate and compare the one LRF based tracking with the two LRFs based one, a second scenario is used. It corresponds to a vehicle which is travelling towards the instrumented one, according to the trajectory illustrated in Figure 5.77.

One can see in Figures 5.78, 5.79, 5.82, 5.84, 5.85, 5.88 and 5.89 that the single LRF based tracking provides bad state estimation, when the vehicle is far. However, the performance of this method increases with the decrease of the distance between the sensor and the tracked vehicle.

The two LRFs based method behaves similarly, but with better vehicle's state estimation. There is, however, a visible difference between the two approaches for distant objects (see Figures 5.80, 5.81, 5.83, 5.86, 5.87, 5.90 and 5.91).

One can see in Figures 5.84 and 5.85 that the IR uncertainty μ_{IR} stays constant at the beginning of the tracking (when the vehicle is far). This is due to the IR line segment length d_{IRx} limitation, as mentioned in section 5.3. In our test, the limit is set to 2 meters.

In Figures 5.78 and 5.79, showing the center position errors for the one LRF based method, one can see great oscillations. This effect is a result of the sensor's low resolution at far distances. To explain the nature of

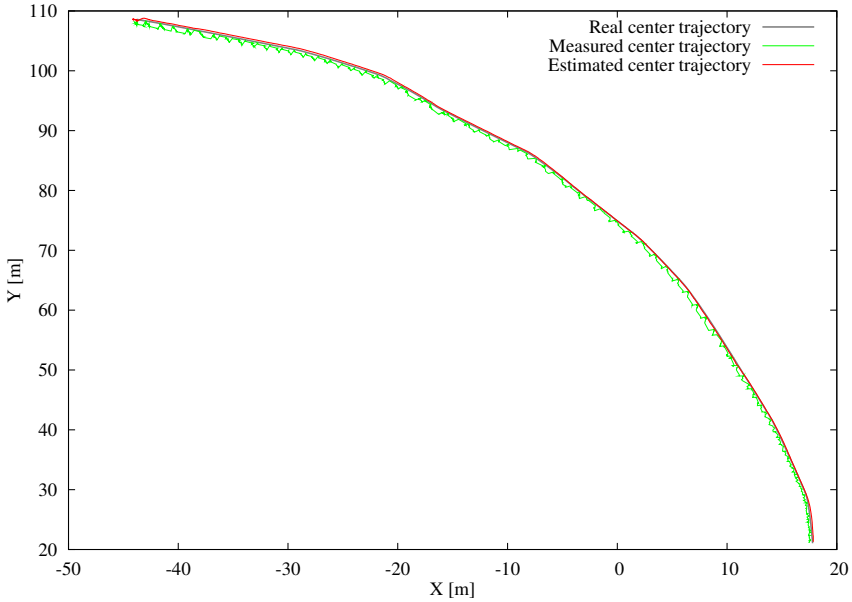

Figure 5.77: Vehicle trajectory (in the instrumented vehicle's local coordinate system).

the problem, let us use the example shown in Figure 5.92. In the example, the real object moves to the right what can be seen as the change of the position in the different time instants. The measurement, however, stays at the same place due to the low laser rays resolution. If the object continues its movement, it will be detected by a new raw data points configuration and, hence, the measurement will change its position. This effect takes place all the time during the tracking of the object. Its intensity is proportional to the laser rays resolution and the velocity of the object. Lower the resolution and the velocity are more prominent the effect becomes, since the time period, when the measurement is static, increases. Thus, in the beginning of the scenario, when the tracked object is far and its speed is low, the object's movement is perceived as a jerking one.

The use of KF smooths the estimated velocity. However, at low speed, when the position of the measurement stays unchanged for a long time, the estimated velocity presents great oscillations. The introduction of the second LRF allows to increase the laser rays resolution and thus the oscillation effect is importantly reduced (see Figures 5.93, 5.94, 5.80, 5.81)

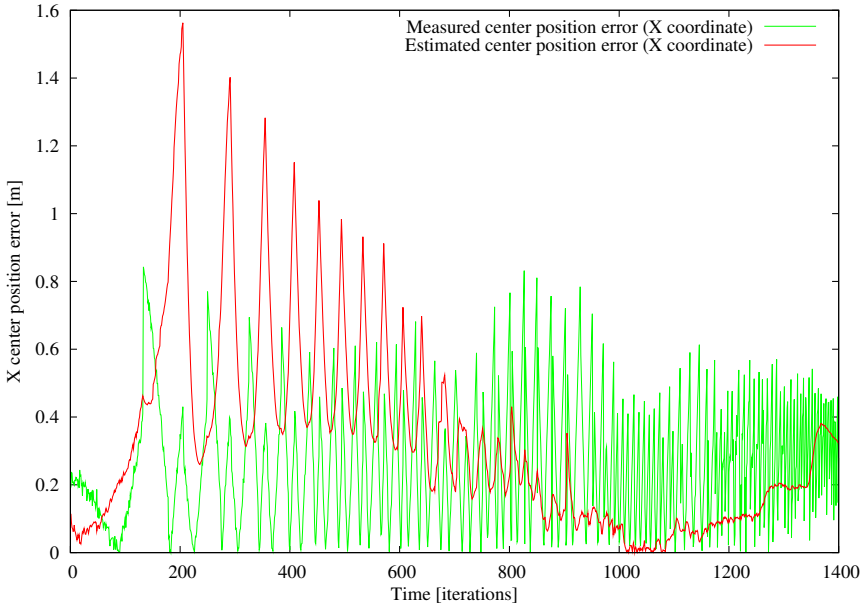

Figure 5.78: One LRF - object's center position error (X coordinate).

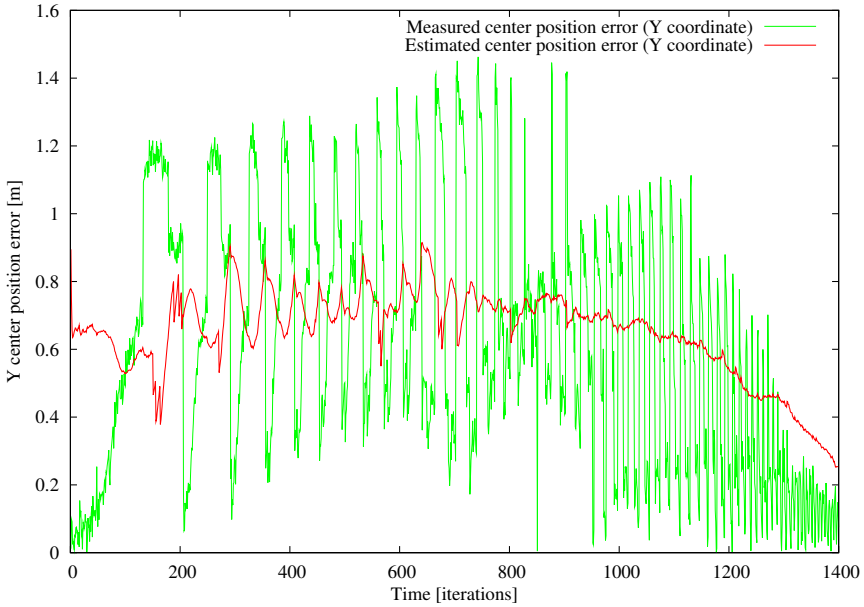

Figure 5.79: One LRF - object's center position error (Y coordinate).

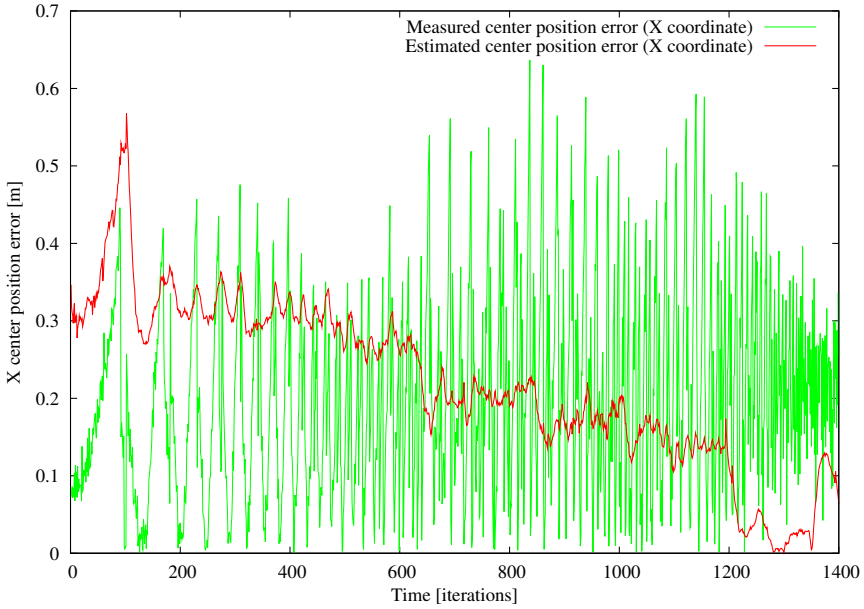

Figure 5.80: Two LRFs - object's center position error (X coordinate).

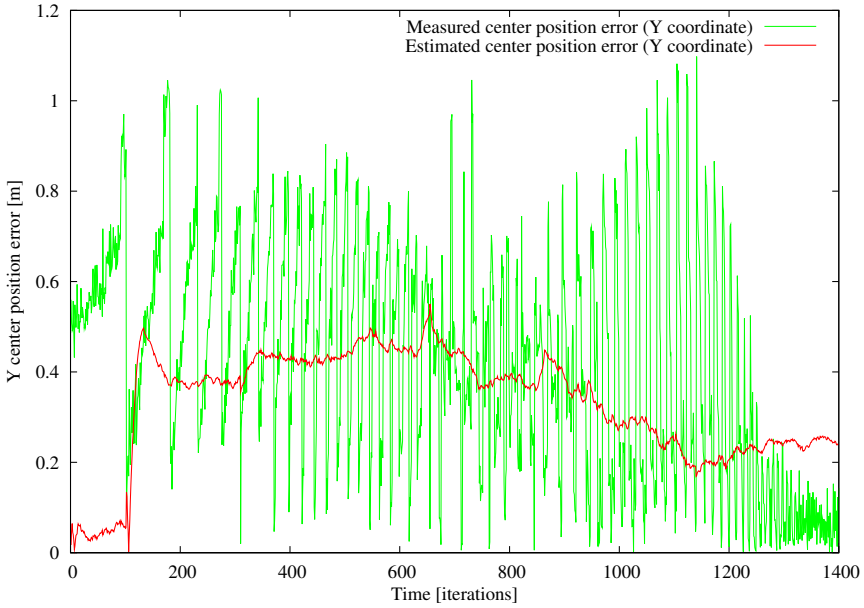

Figure 5.81: Two LRFs - object's center position error (Y coordinate).

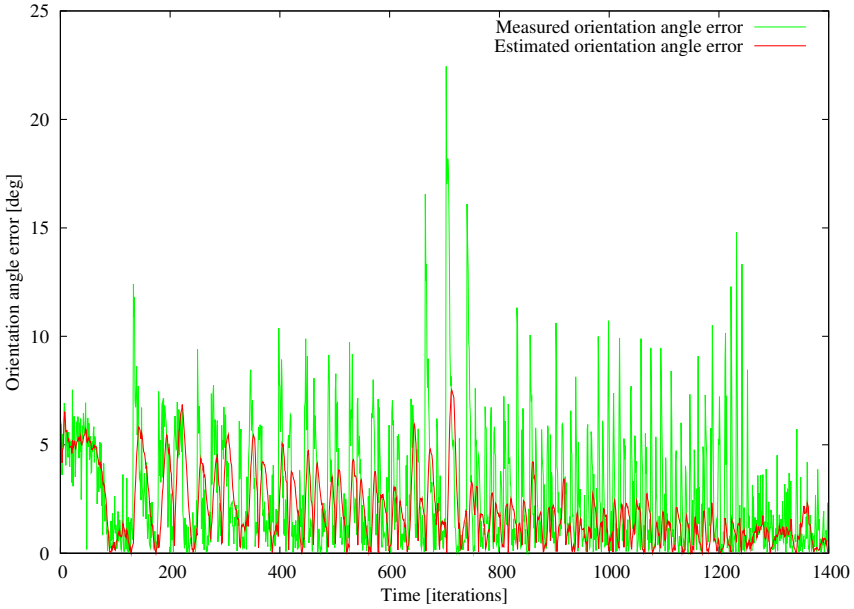

Figure 5.82: One LRF - object's orientation angle error.

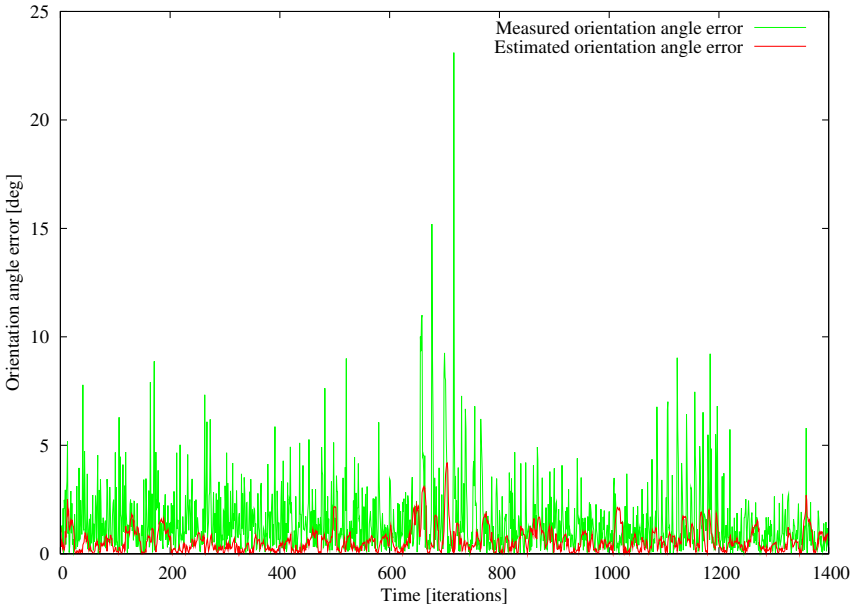

Figure 5.83: Two LRFs - object's orientation angle error.

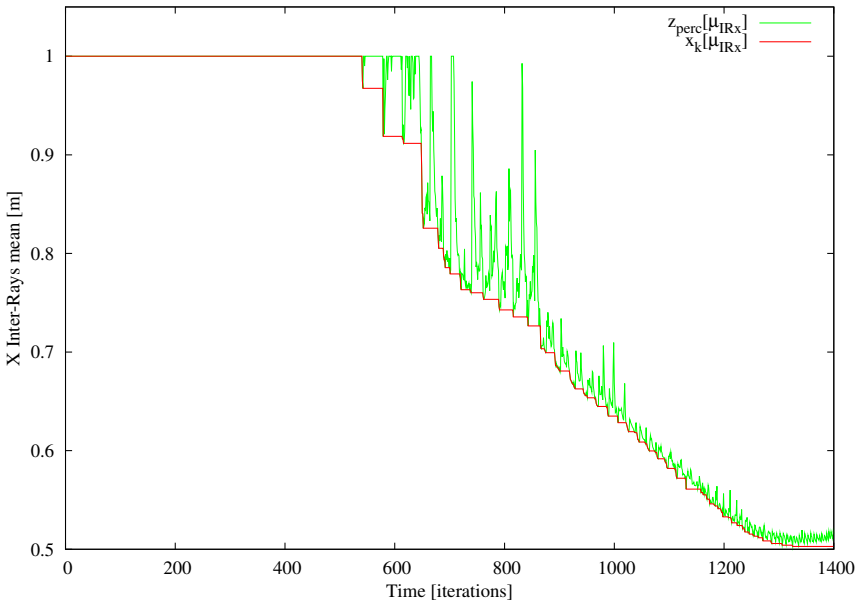

Figure 5.84: One LRF - Inter-Rays uncertainty μ_{IR} (X coordinate).

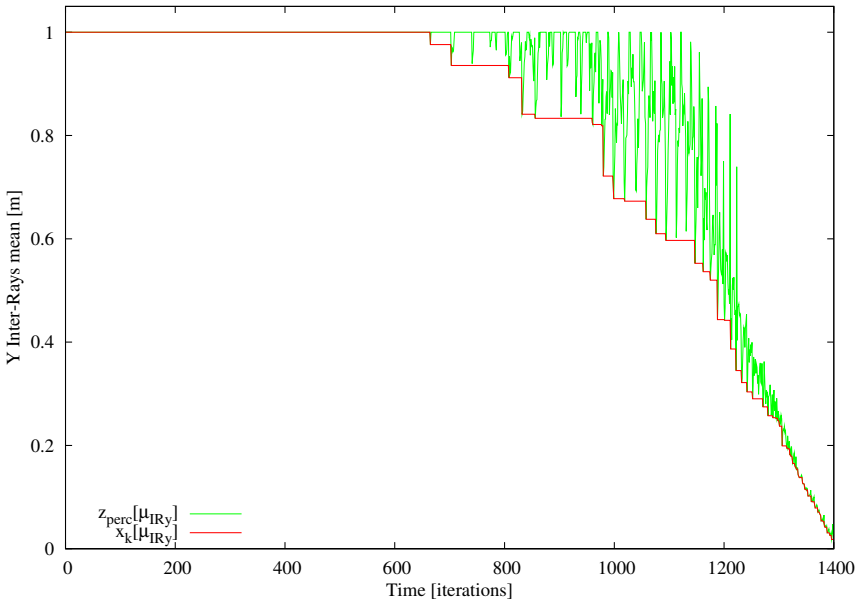

Figure 5.85: One LRF - Inter-Rays uncertainty μ_{IR} (Y coordinate).

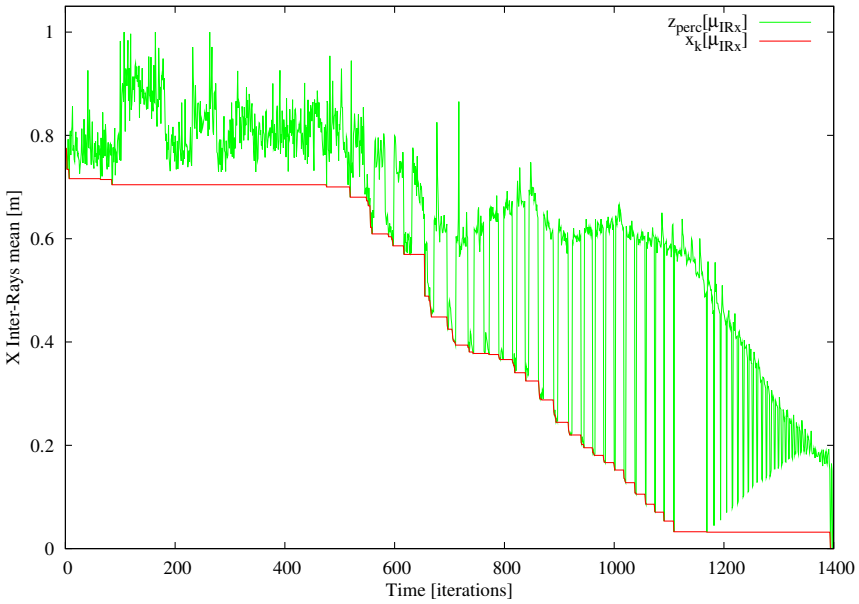

Figure 5.86: Two LRFs - Inter-Rays uncertainty μ_{IR} (X coordinate).

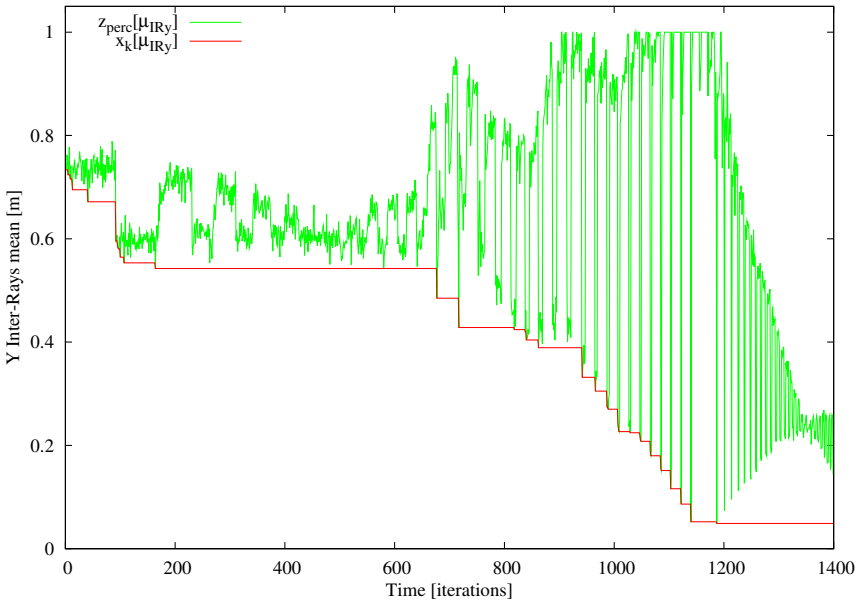

Figure 5.87: Two LRFs - Inter-Rays uncertainty μ_{IR} (Y coordinate).

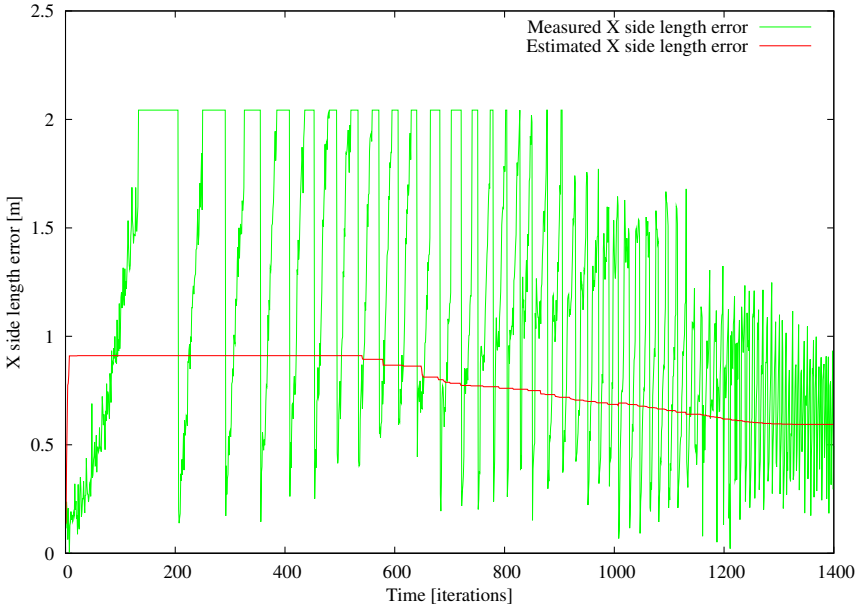

Figure 5.88: One LRF - object's X side size error.

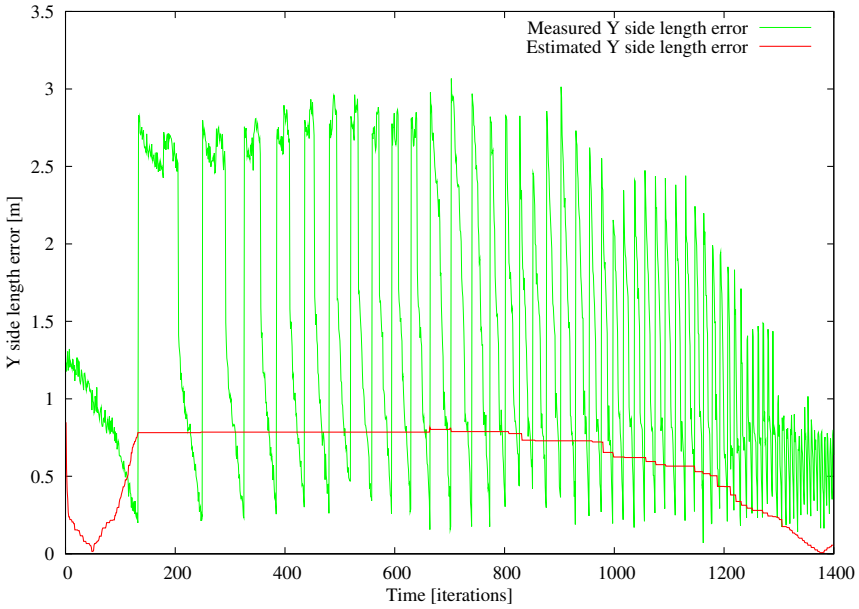

Figure 5.89: One LRF - object's Y side size error.

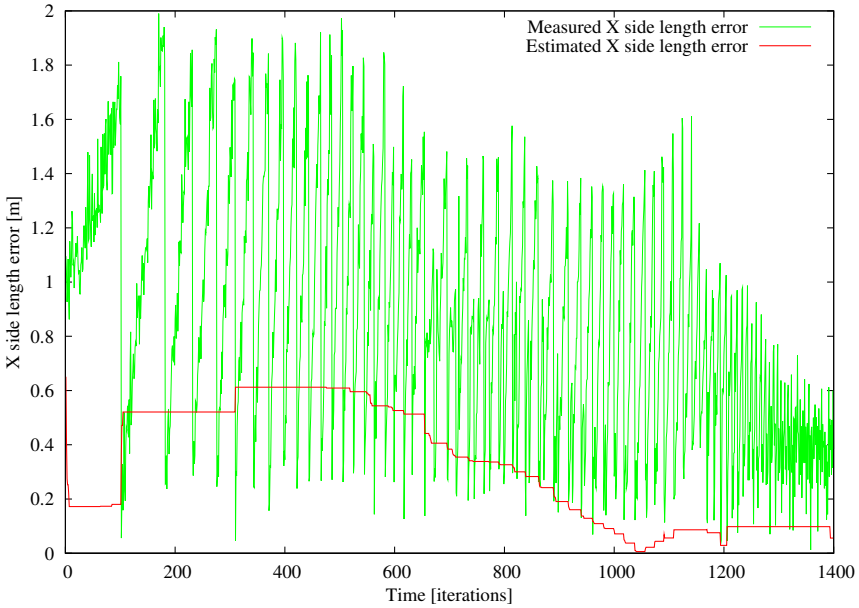

Figure 5.90: Two LRFs - object's X side size error.

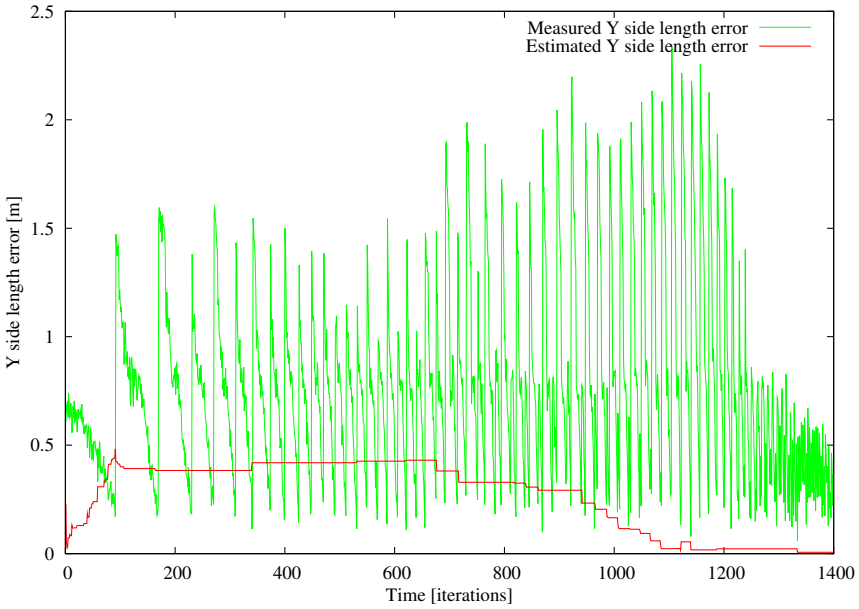

Figure 5.91: Two LRFs - object's Y side size error.

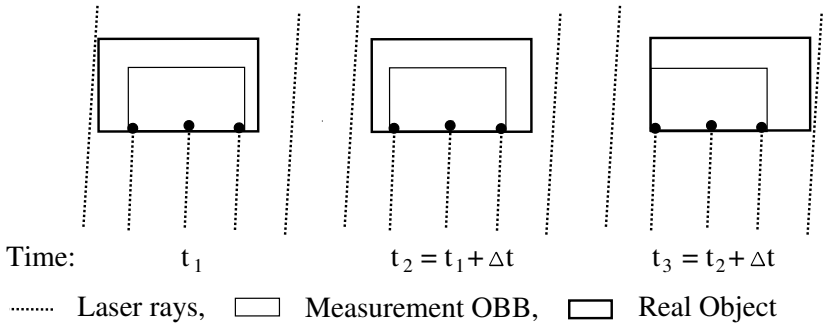

Figure 5.92: Example of the measurement OBB extraction for different object positions for greater distances (small LRF resolutions).

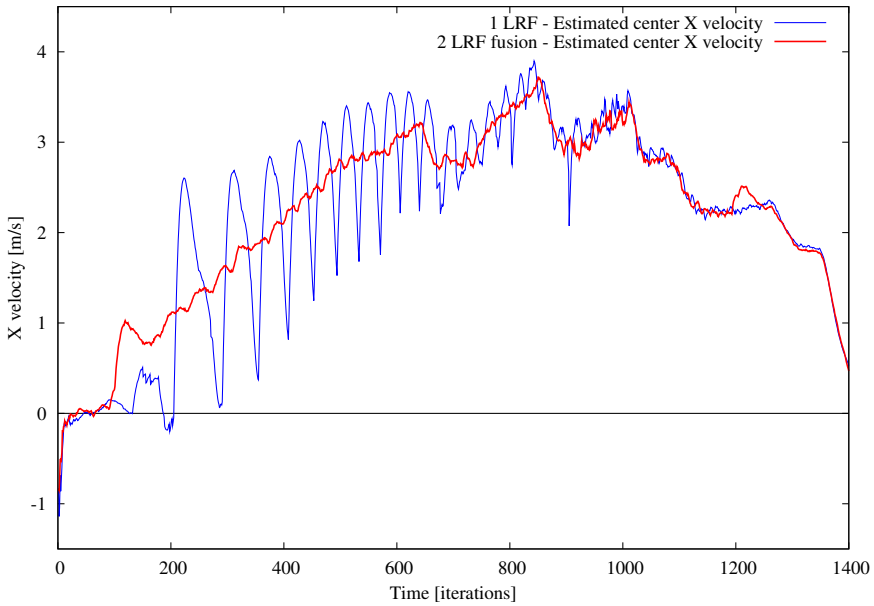

Figure 5.93: Comparison of the velocity estimation between one LRF and two LRF fusion (coordinate X).

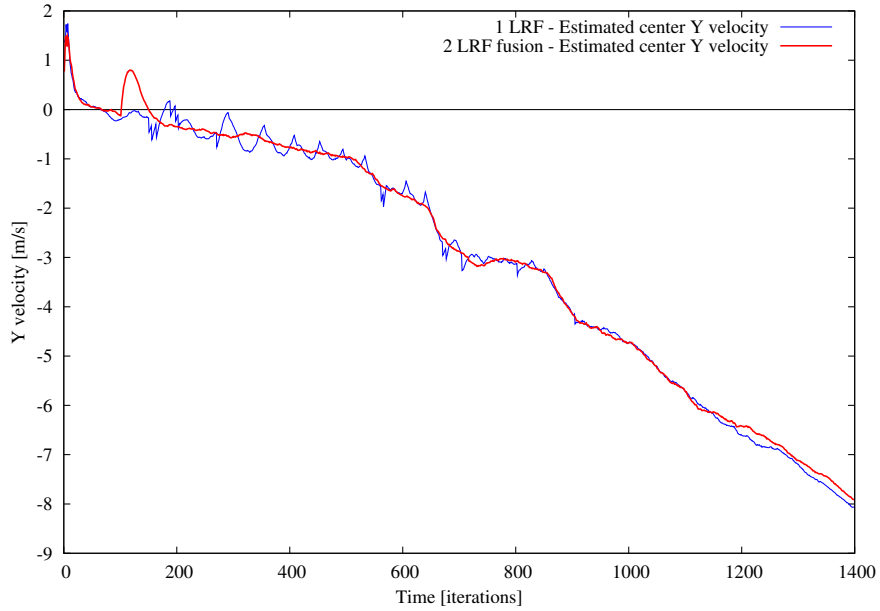

Figure 5.94: Comparison of the velocity estimation between one LRF and two LRF fusion (coordinate Y).

5.6.2 Conclusions

A two-LRF based fusion method for objects tracking is presented. The Oriented Bounding Box model is used to represent the tracked objects. Enriched by the Inter-Rays uncertainty and Fixed Size assumption paradigms, the OBB model performs well with a single LRF, except for far objects, because of the limited angular resolution of the sensor. To overcome this limitation, we have proposed to use two LRF in order to increase the perception angular resolution. The raw data fusion method leads to a better object state estimation. Furthermore, interlacing rays allows additional size estimation refinement using the IR uncertainty. The experimental results have shown the reliability of the two-LRF based fusion system, especially for far objects, when compared with the usage of a single LRF.

Chapter 6

Data association

6.1 Introduction

One of the most important tasks of autonomous navigation in urban areas is tracking of dynamic objects. Data association, which is closely related to the objects representation and sensory data, is a crucial part of the tracking process.

Data association algorithms are composed of the following stages (see Figure 6.1): raw data points clustering, tracks to clusters correlation and raw data points to track association. The first stage is treated as a lightweight preprocessing of data to be associated. The third step consists of more precise, but more time consuming processes, which operate on the first stage's output. The decision of which process will be performed depends on the outcome of the second stage. Tracks, that do not have any raw data points associated, stay valid for the next iteration with the increasing of their age. The tracks, that exceed the maximum live span, are deleted.

The first stage (raw data points clustering) is treated as a preliminary association. It divides the domain of raw data points into subdomains in form of clusters and do not produces precise objects separation. The resulting clusters are then processed by more reliable algorithms of data association. The clustering algorithm is based on a distance threshold. It means that the Euclidean distance between points belonging to the same cluster is below a certain threshold.

The second stage is tracks to clusters correlation. A track is correlated with a cluster if the track's Oriented Bounding Box intersects with the cluster's Axis Aligned Bounding Box. If the track do not intersect any

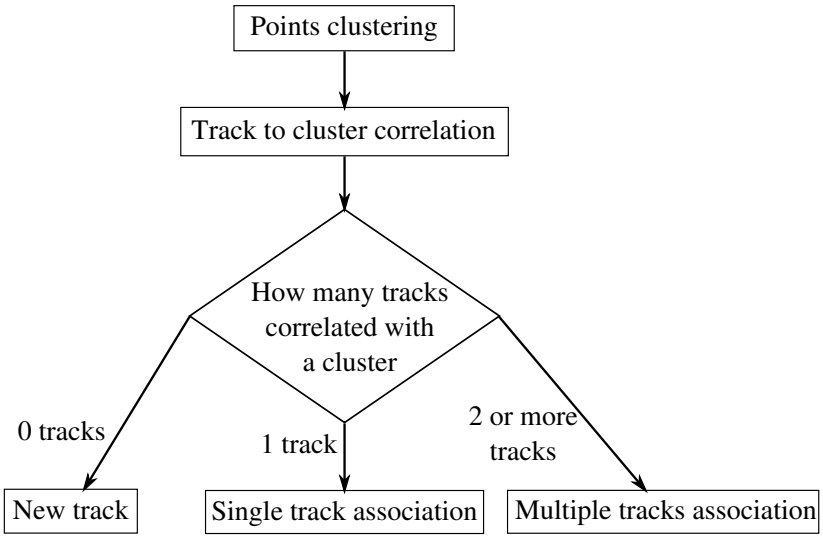

Figure 6.1: Data association flowchart.

cluster, the track is correlated with the closest cluster. There are three possible outputs of tracks to clusters correlation. A cluster can be correlated with zero, one, two or more tracks. These cases represent respectively the following situations: appearance of a new object, tracking of a separated object, and multi-object tracking (see Figure 6.1).

Basing on the results of the previous step, raw data points to track association third stage follows. In this stage, raw data points, positively associated with a track, create a measurement. Each measurement is in the OBB format (see (5.1) and (5.2)).

In the first situation (appearance of a new object), all the points of a cluster are used to create a measurement. In the second one, Mahalanobis distance based gating is used to associate raw data points with a track. Not associated points undergo the local process of clustering and create new tracks.

In the last case, where there are two or more tracks correlated with a cluster, a method based on the Nearest-Neighbour principle is used. The method allows to associate raw data points to existing tracks. Two approaches are proposed. The first one is a simple application of the Nearest-Neighbour principle to the OBB based object representation. The second approach uses the additional information about tracking objects, provided by the Fixed Size (FS) assumption.

As it was already mentioned, in our case, the clustering is only a pre-association stage of the whole association process. To explain in more details

the influence of the clustering step on the general association algorithm, let us define two general cases: an object, which is already tracked and have well defined size; an object, which appears or does not have well defined size. We recall that the FS algorithm is used to estimate and store the size of the objects. In the first case, the clustering process serves only to decrease the data needed to be treated by third stage of the data association algorithm. The this stage can correctly associate points to tracks basing on the tracks state information originating from the previous iteration of the tracking algorithm. In the second case, however, this information is not available, and thus, the application of the third stage of the data association algorithm is useless. This implies that the clustering stage must separate the objects from the raw data points, and, for each cluster, a new track is initialized.

Thresholding used in the clustering is not an adaptive one, and is chosen experimentally. Two different thresholds are used: all points threshold and a consecutive points threshold. The consecutive points are the raw data points produced by neighbouring laser rays. The first threshold is applied for all points, without any relationship condition. The second one is considered only for consecutive points, and is grater then the first one. The introduction of the consecutive points threshold comes from the observation that, very often, consecutive points that do not meet all points threshold, and thus, are put into different clusters, represent the same object (see Figure 6.2).

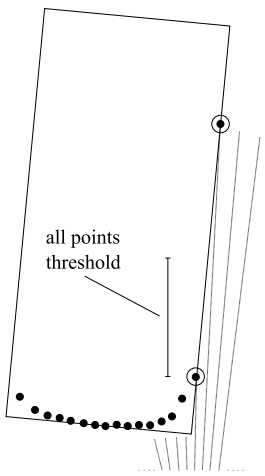

Figure 6.2: Neighbouring points threshold.

The proposed reasoning has, however, a drawback. The problem is to find an appropriate value of the neighboring points threshold. Nevertheless, it is not reliable while used in track initialization. Even if, eventually, the neighboring points threshold has a correct value it cannot give acceptable

results for dynamic obstacles like vehicles while the shorter side of the objects is well visible. Thus, it cannot be applied for general cases. Instead, one can use on of the existing adaptive thresholding methods [49, 21, 65, 17]. The adaptive thresholding allows to cluster the data points more reliably. These methods, however, are based on planar surface assumption. This assumption is relaxed in the proposed earlier OBB based object representation model and extraction algorithm. What is more, these methods are vulnerable to missing or erroneous laser measurements, while the adaptive threshold is applied to consecutive points only. A methods, which uses stereovision information to increase reliability of the laser data points clustering, in this chapter is proposed and thus is resistant to erroneous laser measurements.

6.2 Nearest Neighbors principle with the Fixed Size assumption for data association

In this chapter, an OBB based method for laser scanner data association is presented. The method resolves the problem of coalescing objects separation. The usage of the OBB representation introduces the geometrical aspect for data association. Since the algorithm bases on the geometry and size of the objects it will work reliably for objects being previously recognised as separated ones.

6.2.1 Introduction

There are two variants of the proposed clustering approach. The first one is a direct application of the Nearest-Neighbour (NN) principle to associate each raw data point, included in the processed cluster, with one of the correlated tracks. The difference with the standard NN approach is thatm in this case, the track is assigned to the measurement, instead of assigning the measurement to the track. It means that each point can originate from only one track, but track can have many measurements (raw data points). To increase the NN algorithm reliability, the second variant of the proposed clustering approach takes into account the object size. The objects size is available if the FS assumption is used. The FS assumption allows to store the best size estimation obtained during the tracking. The robustness of the second variant lies in the correctness of the object size estimation. In the proposed approach the problem of occlusion is not taken into account.

This is due to the fact that the FS algorithm, in the case when an object with only one visible side is occluded, gives wrong position estimation.

6.2.2 Nearest-Neighbour variant

For each pair of a raw data point and an OBB, representing a track correlated with the cluster, Mahalanobis distance is calculated. If the raw data point is validated by only one track gate, it is assigned with that track. If the raw data point is situated in the gates of a few tracks, it is assigned with the closest track, in Mahalanobis distance sense.

The size of the gate is related to the track position prediction covariance by applying the following gate rule:

$$T_{d^2} \geq d^2_{(ij)} = \nu^T_{(ij)} S^{-1} \nu_{(ij)} \tag{6.1}$$

where $\nu_{(ij)}$ is the Euclidean distance vector defined from the ith raw data point to the jth track's prediction OBB, S^{-1} is the inverse of the track position prediction covariance matrix. A value of the gate's threshold T_{d^2} is taken from χ^2 distribution with two degrees of freedom, and expresses the number of sigmas of the Normal distribution. The track position prediction covariance matrix S is constructed as follows :

$$S = \begin{bmatrix} \sigma^{-2}_{cx} & 0 \\ 0 & \sigma^{-2}_{cy} \end{bmatrix} \tag{6.2}$$

where σ^{-2}_{cx}, σ^{-2}_{cy} are the OBB center position prediction uncertainties in X and Y coordinate.

The advantage of this method is simplicity and low computational complexity. The method, however, does not always give reliable results. It is stable for the cases where the distance between tracked objects is greater than the objects gates. The object gate is proportional to the object position prediction uncertainty, which depends on the sensor's range uncertainty and the object displacement between sensor readouts. Thus, the smaller distances between the tracked objects are and the greater the displacement is, the more probable the approach will fail.

6.2.3 Nearest-Neighbour with Fixed Size variant

The second variant combines the Nearest Neighbors principle with the Fixed Size assumption (NN+FS). The Fixed Size assumption is used to improve the NN association algorithm. Figure 6.3 shows the schema of the NN+FS

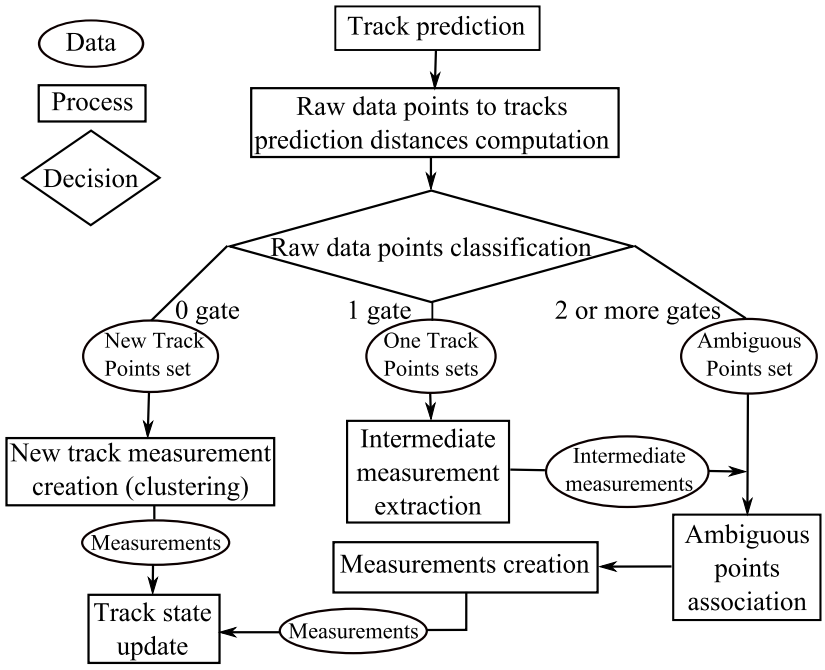

Figure 6.3: Schema of NN+FS data association algorithm.

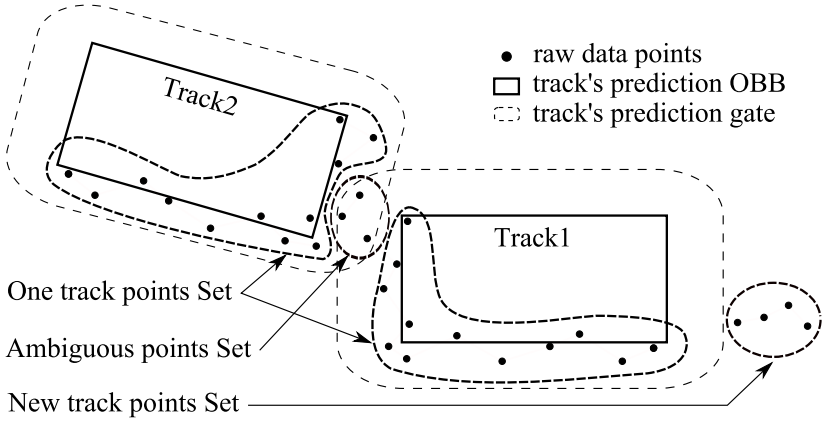

Figure 6.4: Classification of raw data points using three classes.

variant. In the first stage of the algorithm, raw data points are classified into three classes: New Track Points (NTP), One Track Points (OTP), Ambiguous Points (AP). The classification process is based on the relation between raw data points and the gates of the tracks in the cluster. The first class represents the points which are outside all the gates. The second class consists of sets of points, which are inside only one track (one set per one track). The last class consists of sets with points, which are inside more than one gate (see Figure 6.4).

The points from the first class (NTP) are separated into clusters (using the same clustering method as in the preliminary association). For each cluster, a new track is created.

Each OTP class set, associated with a track, serves as a source for obtaining intermediate measurement. In this process, the basic OBB extraction method, presented in section 5.1, is used.

The aim of the last stage of the proposed NN+FS algorithm is to tackle ambiguous points association using the AP class set and intermediate measurements. Each ambiguous point $P_{(i)}$ in this set has a list $L_{(i)} = \{x_{(j)}\}$ of tracks to which it may belong (point is inside the gate of each track of the list), where $i \in [1; N]$ is a point number, N is the AP set cardinality, $x_{(j)}$ is the jth track. For each pair composed by a point $P_{(i)}$ and a track $x_{(j)}$ from the list $L_{(i)}$, the hypothesis $H_{P_{(i)},x_{(j)}}$ that the ith point originates from the jth track is tested. For this point-track pair $(P_{(i)}, x_{(j)})$, we create a temporary OBB $z_{(ij)temp}$, constructed by including the point in the track's intermediate measurement. If the temporary OBB size is not greater than the jth track prediction OBB size, the point can be associated with the jth track. Otherwise, the point is not associated with the track. It happens that the point $P_{(i)}$ is associated with more than one track. In this case, this point is associated with the jth track for which the difference:

$$Diff_{(ij)} = Diff_{(ij)}[dx] \cdot Diff_{(ij)}[dy] \tag{6.3}$$

is the smallest, where:

$$Diff_{(ij)}[dx] = |z_{(ij)temp}[dx] - \hat{x}_{(j)}(t|t-1)[dx]| \tag{6.4}$$

and

$$Diff_{(ij)}[dy] = |z_{(ij)temp}[dy] - \hat{x}_{(j)}(t|t-1)[dy]|. \tag{6.5}$$

6.2.4 Simulator based results

For the test of the proposed algorithms, a Laser Range Finder (LRF) is mounted in front of the vehicle. The step angle for the LRF is set to 1° with an angle range of 180°. In the tests, the sensor range uncertainty σ_ρ is set to 0.05 m.

The proposed algorithm is evaluated using two scenarios, with two tracked vehicles (see Figure 6.5). The scenarios are chosen to show the reliability of the proposed algorithm, which stay stable even when two objects touch themselves.

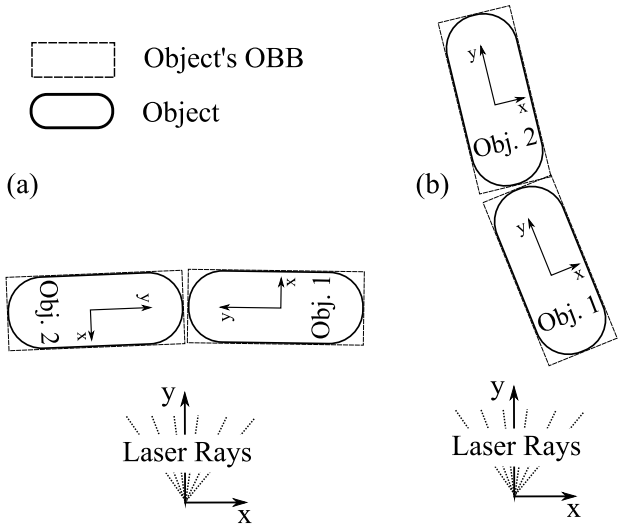

Figure 6.5: Objects configurations during the phase of "multi-track data association" (a) First scenario, (b) second scenario.

In the first scenario, two vehicles run towards each other by traveling a symmetrical trajectory with respect to the Y axis of the LRF reference. In the moment of the vehicles frontal position, the angles between the vehicles' Y sides and the intersecting LRF rays are close to the right angle. Thus, only the Y side of the vehicles is seen by the LRF.

In the second scenario, the first vehicle runs towards the second one, which does not move. When the two vehicles become close to each other, the angles between the vehicles' Y sides and the LRF rays are very small. Furthermore, the X side of the second vehicle becomes occluded by the first vehicle.

In the second scenario, because of the vehicles orientation, the LRF range uncertainty makes data association more difficult than in the first one.

We can see at the end of the two scenarios that the two vehicles collide, and one vehicle pushes the other one (see Figure 6.5). This part of the two scenario is considered to show that NN+FS data association algorithm remains reliable even in this extreme situation.

Three approaches are evaluated. The first one is a pure NN based algorithm with the EKF based filtering. The second approach is the a NN based algorithm with the EKF based filtering using the IR uncertainty and the FS assumption. The third approach is a NN+FS (NN enriched by track's size information) based algorithm with the EKF based filtering using the IR uncertainty and the FS assumption.

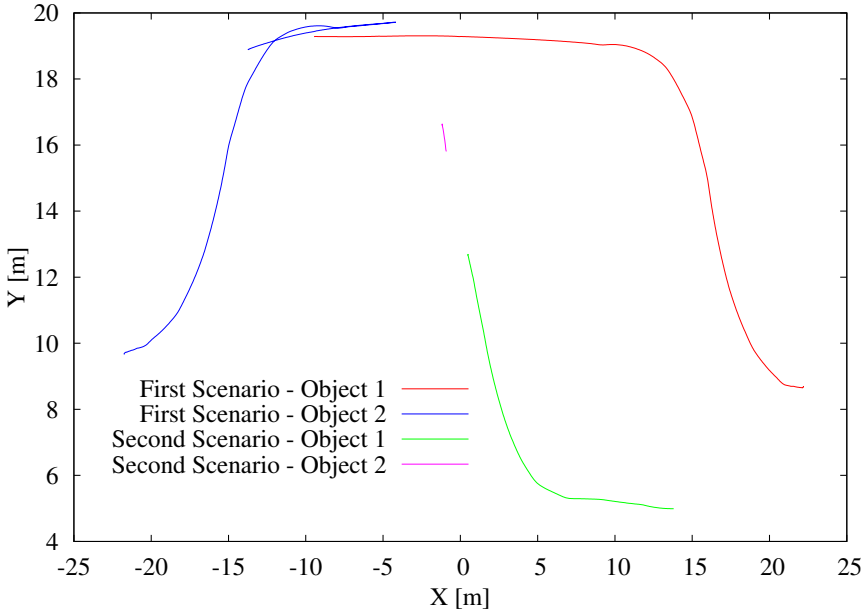

Figure 6.6: Trajectories of the tracked vehicles for both scenarios.

To evaluate the approaches, the vehicles' size estimation (Y side only - in the local tracks coordination system) is compared. Size information gives the best insight into the performance of the tested approaches, since badly associated points influence directly the track's size. In the proposed scenarios, the Y side of the vehicles is visible during the phase of "multi-track data association" (see Figure 6.5), while the X side is not visible or occluded, and so, cannot be used for comparison. The estimated trajectories of the tracked vehicles are presented in figure 6.7.

In figures 6.7-6.12, one can see the real object Y side size (the two tracked objects are identical in terms of size) and its estimation for each vehicle.

Considering the pure NN algorithm (see Figures 6.7, 6.10), the absence of the IR uncertainty leads to underestimated object size. Furthermore, the absence of the FS assumption does not guarantee that the object size decreases in time (what is different with reality). The integration of the IR uncertainty allows a better estimation of the objects size. The correct size estimation is assured by using the FS assumption, despite unfavourable position and/or orientation of the objects (see Figures 6.8, 6.9, 6.11 and 6.12.

Figures 6.7 - 6.9 show the evolution of the vehicles' size estimation (Y side only - in the local tracks coordination system) in the first scenario. The "multi-track data association" phase starts after about 500 iterations. One can see that the first approach (pure NN) fails, the objects' size estimates get worse with time. The points originally belonging to the track number

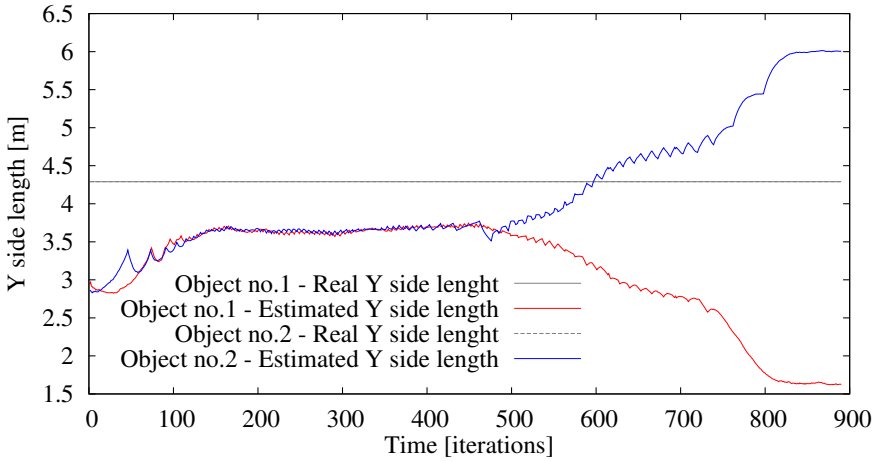

Figure 6.7: First scenario - Evolution of the vehicles' size (Y side) using NN method without IR and FS.

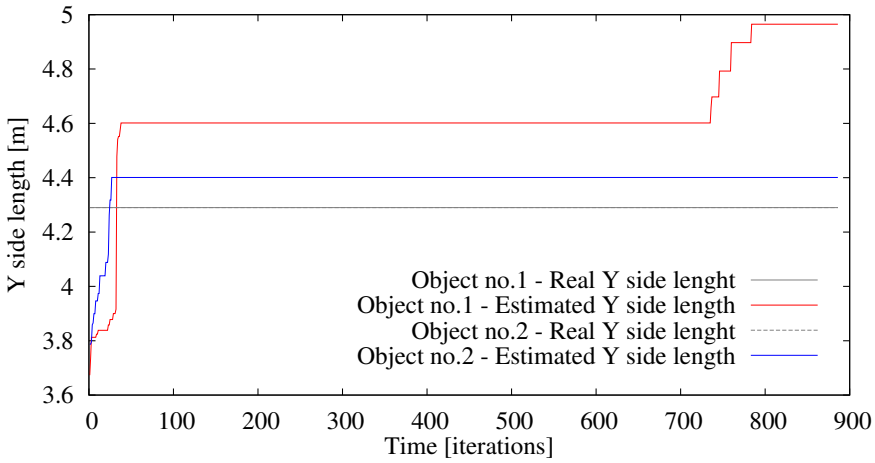

Figure 6.8: First scenario - Evolution of the vehicles' size (Y side) using NN method with IR and FS based tracking.

1 are associated with the track number 2. The second approach (pure NN with IR and FS based tracking) stays stable for a certain period, but finally it also fails. The result is similar to the results of the first approach. Indeed, the points originally belonging to the track number 2 are associated with the track number 1, with the difference that the second track size stays constant due to the FS assumption. Only the third approach (NN+FS)

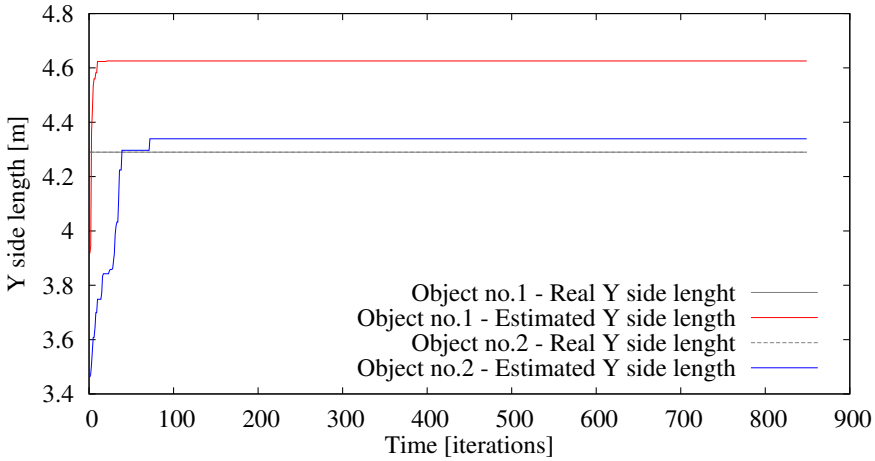

Figure 6.9: First scenario - Evolution of the vehicles' size (Y side) using NN+FS method.

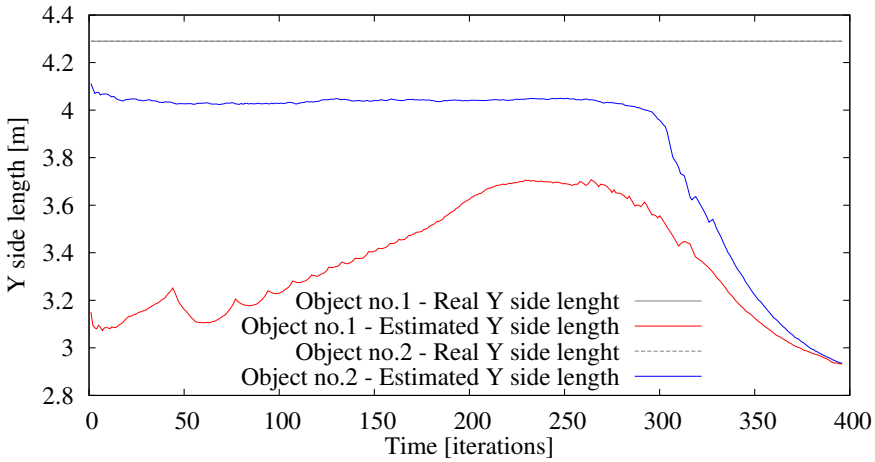

Figure 6.10: Second scenario - Evolution of the vehicles' size (Y side) using NN method without IR and FS.

stays stable and manages well to correctly associate points to tracks.

Figures 6.10 - 6.12 show the evolution of the vehicles' size estimation (Y side only - in the local tracks coordination system) in the second scenario. The "multi-track data association" phase starts after about 300 iterations.

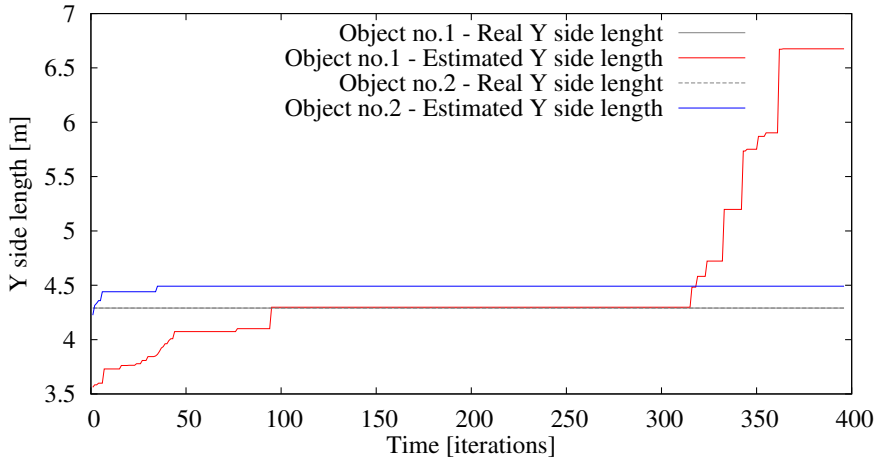

Figure 6.11: Second scenario - Evolution of the vehicles' size (Y side) using NN method with IR and FS based tracking.

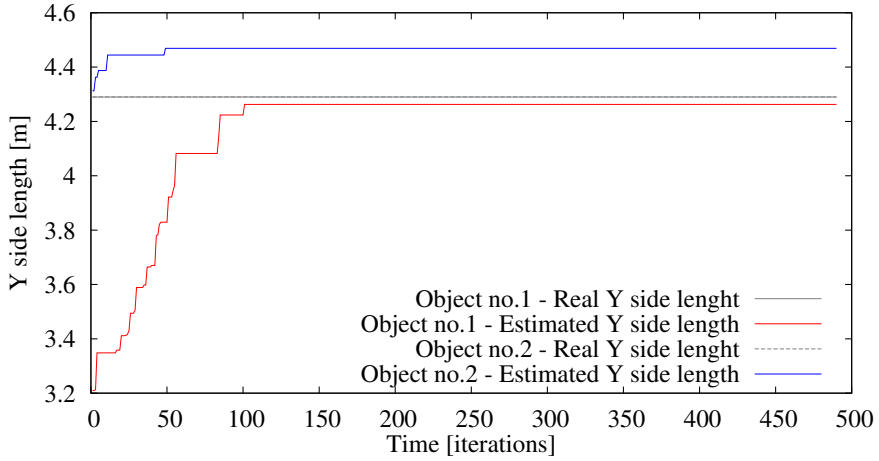

Figure 6.12: Second scenario - Evolution of the vehicles' size (Y side) using NN+FS method.

In this scenario, the first approach (pure NN) manages to give the correct data association. This is due to the favourable raw data points configuration (see Figure 6.10). However, one can see that the sizes are greatly underestimated due to the absence of the IR uncertainty and the FS assumption in the tracking.

The second approach (pure NN with IR and FS based tracking) fails and the technique performs as in the first scenario (see Figure 6.11). The third approach (NN+FS) performs well and all points are correctly associated (see Figure 6.12).

6.2.5 Real platform based results

The comparison of simulator based results shows that the NN+FS approach is the most reliable of the three tested ones. That is why, in real conditions, we confirm only NN+FS approach. The real data based evaluation consists of two scenarios. The first one is similar to the first scenario used in the simulator based evaluation (see Figures 6.5(b) and 6.13). One of the vehicles approaches to the first stationary one. It stops when the distance between the vehicles is very small. After a few moments both vehicles move forwards. Figure 6.18 presents the center trajectory of both cars. In figures 6.15 and 6.17, the evolution of the X and Y sides size of the vehicles are presented. The results show, that the method is stable and associates correctly the laser points. This can be concluded from the sides sizes, which does not change during the maneuver.

Figure 6.13: First scenario - image sequence

In the second used scenario, one of the vehicles passes close to the second one, which is stationary. The passage is repeated two times: from near to far and from far to near (see Figure 6.17). In figure 6.18, the estimated object trajectories of both vehicles are presented. The vehicle number one is stationary during the experiment, thus, the trajectory is visible as a small mark. From figures 6.19 and 6.20, one can see that the estimated size of the two vehicles does not change what confirms the reliability of the proposed method. In this scenario, the Y side of the first vehicle is not visible, thus, its size estimation is very underestimated (see Figure 6.20)

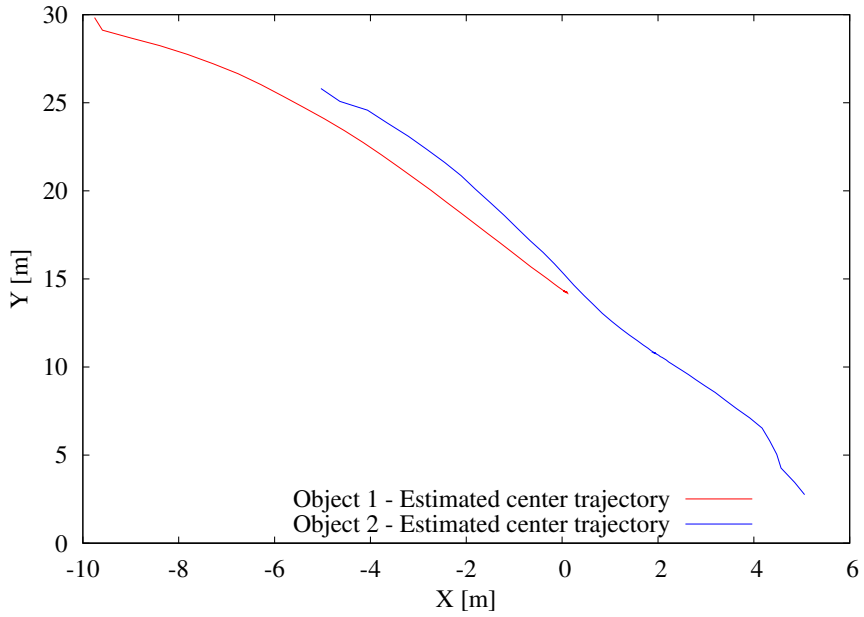

Figure 6.14: First scenario - vehicles' trajectories

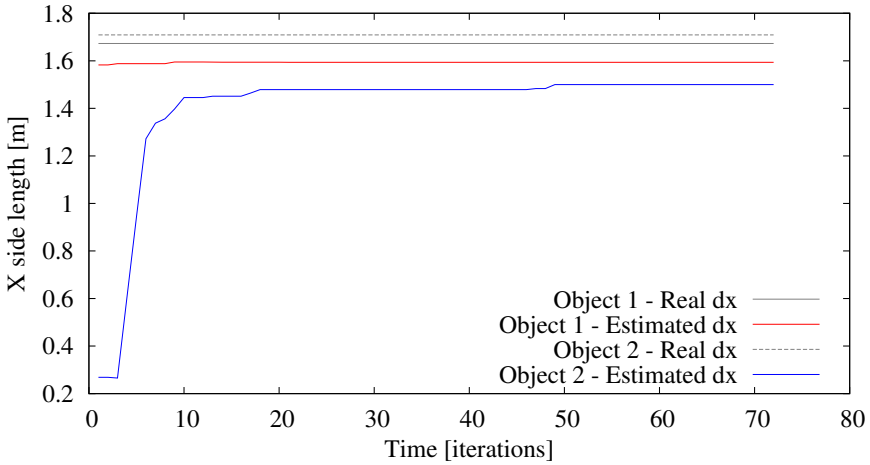

Figure 6.15: First scenario - Evolution of the vehicles' size (X side) using the NN method with the IR and FS based tracking.

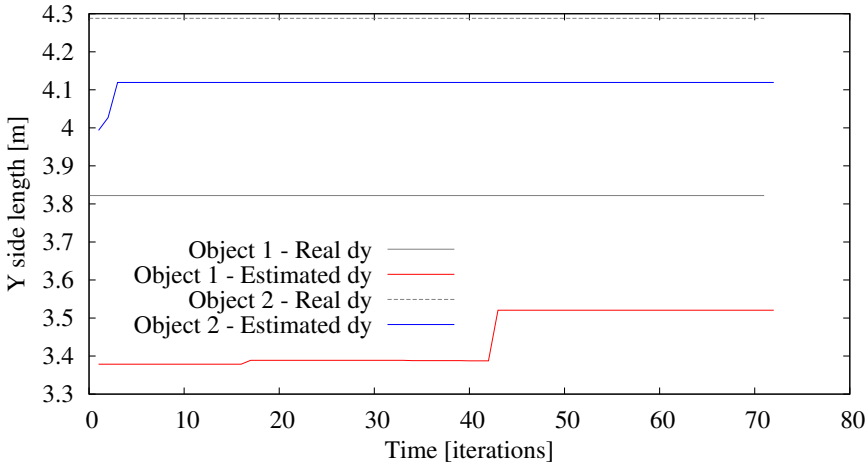

Figure 6.16: First scenario - Evolution of the vehicles' size (Y side) using the NN method with the IR and FS based tracking.

Figure 6.17: Second scenario - image sequence

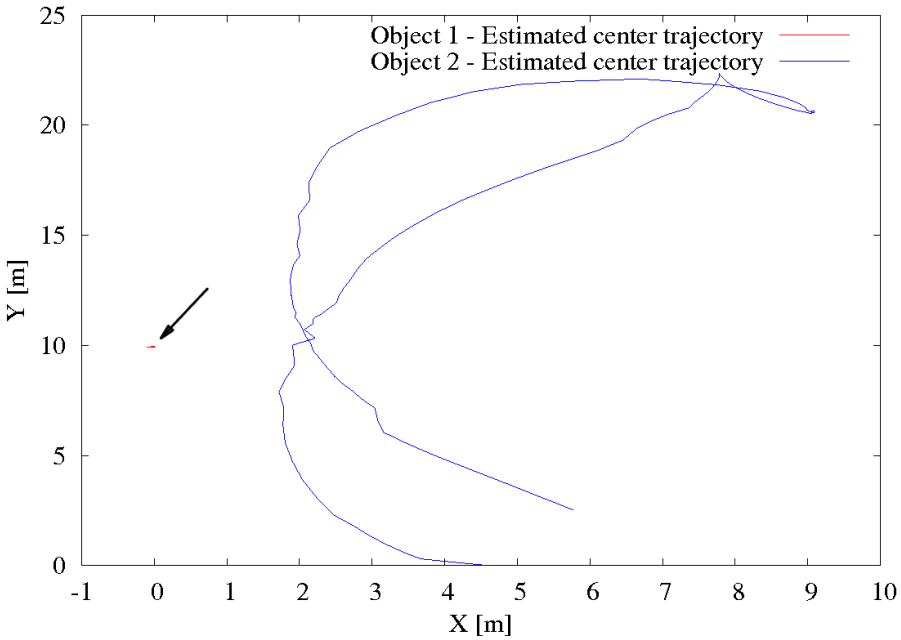

Figure 6.18: First scenario - vehicles' trajectories

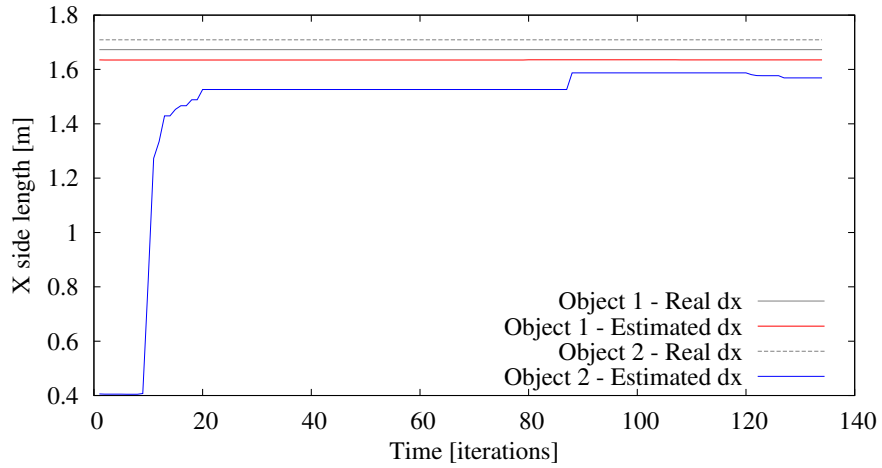

Figure 6.19: Second scenario - Evolution of the vehicles' size (X side) using the NN method with the IR and FS based tracking.

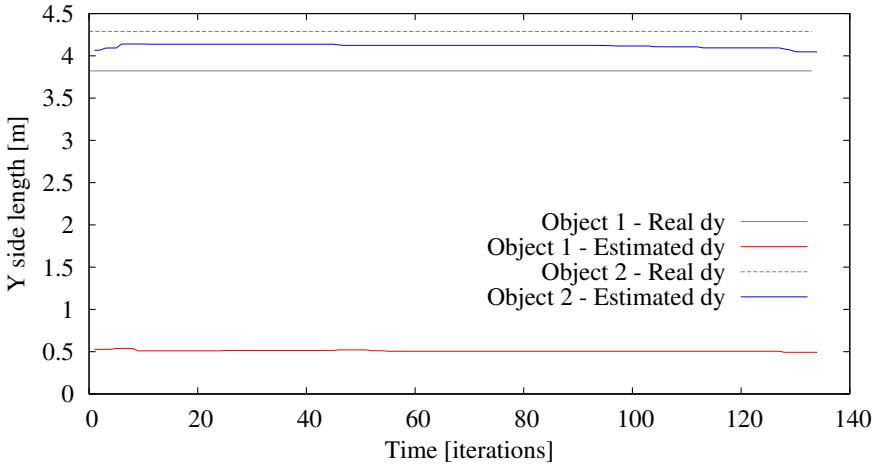

Figure 6.20: Second scenario - Evolution of the vehicles' size (Y side) using the NN method with the IR and FS based tracking.

6.2.6 Conclusions

In this section, two methods for raw data points association to track were presented. To evaluate the proposed methods, three approaches were compared: NN for OBB without IR and FS, NN for OBB with FS and NN+FS. The analysis, based on simulated data, is done with different scenarios to take into account different objects special configurations. The experimental results show that the NN+FS method performs more reliably than the other tested approaches. The robustness of the NN+FS method was confirmed by evaluating the method on real data.

6.3 LRF and Stereovision fusion for raw data points clustering

6.3.1 Introduction

In section 6.1, it was pointed out that preliminary raw data point clustering has some drawbacks, and does not give correct results for certain situations. The first problem is that there is no optimal neighboring points threshold value for unstructured environments such as urban areas. Additionally, it cannot be used for track initialization and to solve general cases. It was mentioned that, in problematic raw data points configurations, it is better, for more reliability, to leave the decision on points clustering to the third stage of the association schema. Adaptive thresholding can be used (see Section2.1.1). Thresholding methods, however, assume that only consecutive scan points can create one surface. In real situations, this assumption may not be met due to erroneous laser readouts.

It is shown, in the last section, that the NN+FS association method gives good results for object having well estimated size. The good size estimation is possible for objects, which are already correctly separated and which have at least both sides were visible. Without LRF and Stereovision based approach, an object presented below can be recognized as separate one if its distance from other objects is greater that threshold used for clustering.

Nevertheless, there are situations where, without additional information, it is impossible to cluster raw data points correctly. One of these situations is illustrated in Figure 6.21. The configuration (a) represents the raw data points corresponding to the rear of a vehicle, seen by the LRF. The configuration (b) could correspond to two possible situations. The first one "vehicle turning" (c) represents the vehicle, which is turning to the right from its former position (a). The second situation "vehicle occlusion" (d) represents two vehicles: the first one (a) and a second vehicle, which is perceived partially by the LRF.

Using only LRF data threshold based the clustering, it is impossible to achieve a correct discrimination between different situations. To discard the ambiguities, it is proposed to fuse LRF data and stereovision information. To allow real-time algorithm execution, the stereovision information is produced and analyzed only within region of interests (ROI). ROIs are defined by projections into the stereo images of the consecutive laser points, which verify a distance constraint. In other terms, for each two neighbouring laser

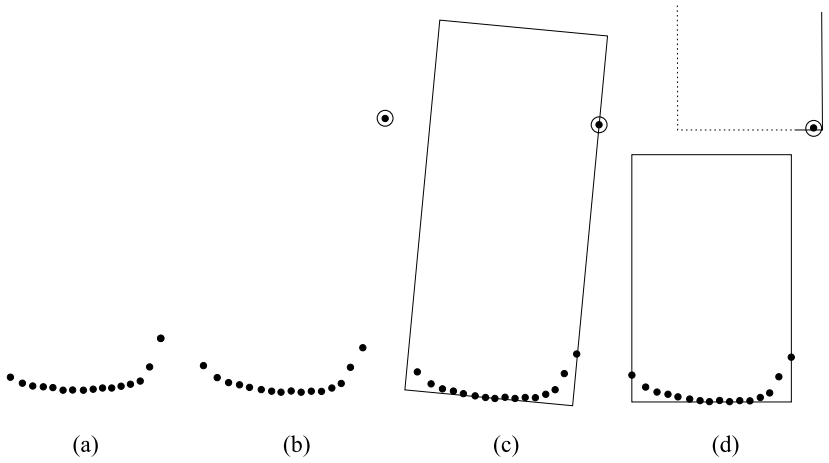

Figure 6.21: Laser data clustering ambiguities.

rays points, for which the distance between them is greater than a certain threshold, the stereovision analysis is performed. The analysis answers the question if the points belong to the same object or not. Figure 6.21 shows a pair of consecutive points (surrounded by circles) for which the stereovision analysis is necessary.

6.3.2 Clustering algorithm

The proposed clustering algorithm is illustrated in Figure 6.22. The schema shows the clustering process for each input point P_j. Input points are processed consecutively. In the first step, a classical threshold based clustering is performed. If the point P_j is not assigned to any cluster, the existence of a consecutive point P_i is checked. If the test fails, the point P_j creates a new cluster, otherwise a process of point gating is lunched. If the point P_j is inside of a track's gate, it is added to the existing track's cluster, or creates a new cluster for that track. If the point P_j is outside of all existing tracks' gates, the stereovision analysis is performed for the points P_j and P_i. Basing on the stereovision analysis, the point P_j creates a new cluster or is added to the cluster of the point P_i.

6.3.3 Disparity map construction

The most difficult part of the disparity map construction is correlation of pixels in stereoscopic images. Depth information is extracted form matched

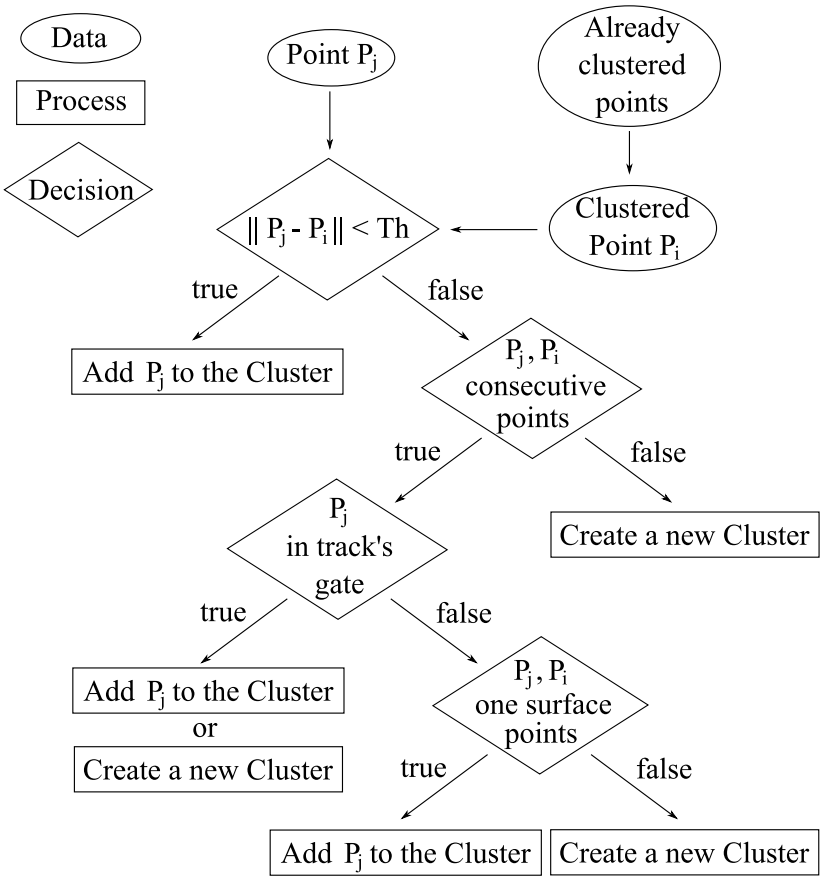

Figure 6.22: Clustering schema.

pixels using geometrical triangulation.

Stereo correlation consists of matching primitives extracted from the pair of stereoscopic images (left and right images). To remove a large number of matching ambiguities, a certain number of local and global constraints are used : epipolar constraint, orientation constraint, constraint of minimal and maximal disparity, uniqueness constraint, ordering constraint, constraint of continuity of the disparity, etc. There are many methods of correlation in the literature [19]. The SAD algorithm (Sum of Absolute Differences) is one of the most popular correlation methods. It is generally used with grayscale images [22, 35, 45]:

$$SAD(f_l, f_r) = D(f_l, f_r) = \|f_l - f_r\| \tag{6.6}$$

where f_l and f_r denote respectively the vectors containing the pixel values of the correlation window in the left and right images.

In the context of road environments, brightness changes in the stereo

images point out some problems during the matching process. To reduce this effect, we use the ZSAD (Zero mean Sum of Absolute Differences) method [19], which is less sensitive to illumination changes. Based on pattern comparison, the ZSAD algorithm is an extension of the SAD algorithm. For each pixel of the image, a correlation window containing the neighbourhood of the considered pixel is used to compare the similarity. For each correlation window in the left image, the correlation window with the greatest similarity is sought in the right image:

$$ZSAD(f_l, f_r) = \left\| (f_l - \overline{f_l}) - (f_r - \overline{f_r}) \right\| \tag{6.7}$$

where f_l and f_r denote respectively the vectors containing the pixel values of the correlation window in the left and right images. $\overline{f_l}$ and $\overline{f_r}$ correspond to the average of the pixel values of f_l and f_r vectors, respectively.

Figure 6.23 shows the two clustering ambiguous situations (previously described; see Figure 6.21), where stereovision information is useful to perform a correct laser data clustering. The first situation concerns a vehicle, which is turning to the right (top-left image). The second one concerns a vehicle partially occluded by another (top-right image). The corresponding disparity maps in grayscales are presented (middle images). In order to visualise better the detected regions, labelled images are extracted (bottom images) from the disparity maps. In these images, each detected region corresponding to a disparity value is represented by a different color.

It is important to notice that for the clustering task, the stereovision analysis is performed only in regions of interest (ROI), defined from the image-projections of pair of consecutive laser points that respect a distance thresholding rule (see section 6.3.4). This feature allows to work with high resolution images in real time. The algorithm is tested with images of resolution 1280 x 960. Considering the example of Figure 6.23, the ROIs are defined from the image-projections of the pairs of the consecutive laser points, surrounded by circles.

Figures 6.24 shows two surface maps of two zones corresponding respectively to the two situations presented in Figure 6.23. The zones are defined by the first and last laser points projected onto the images.

Each surface map is represented by the disparity values (Z-axis) within the image space (X-axis and Y-axis). Farther is the pixel, smaller is the disparity value (yellow to purple).

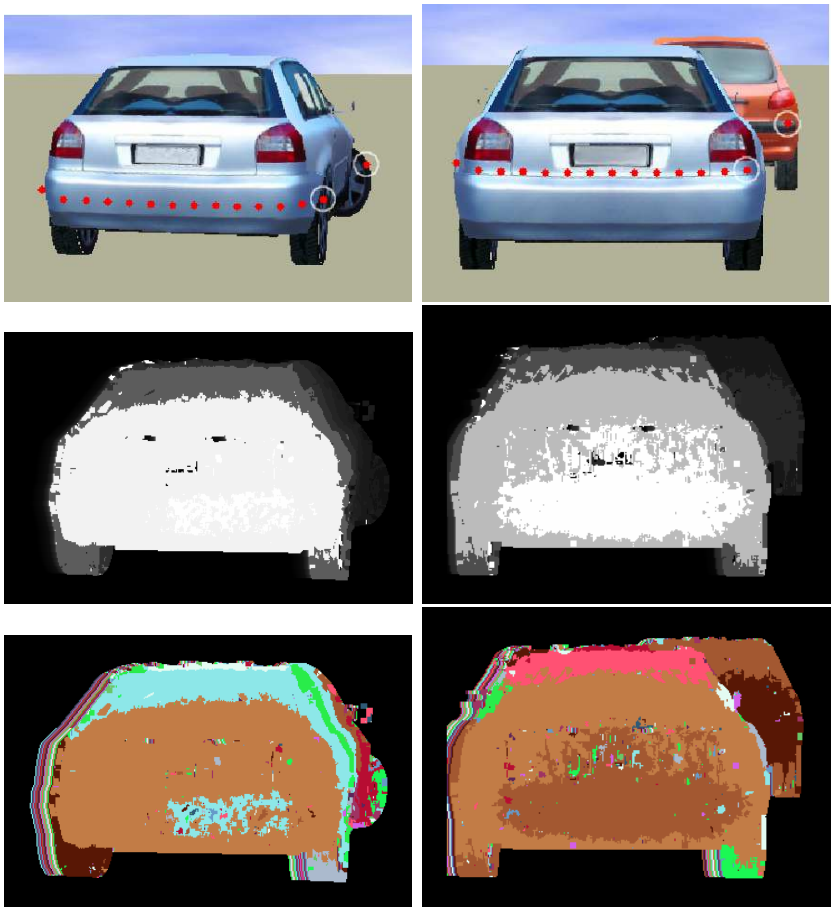

Figure 6.23: Disparity maps original images (top), their disparity maps in grayscales (middle) with labelled images (bottom) for the "vehicle turning" (left) and the "vehicle occlusion" (right) clustering ambiguous situation.

In the fist scenario (the vehicle turning to the right), one can notice that the disparity values are globally constant in the left part. In the right part, the values decrease gradually. This part corresponds to the ROI defined by the two projected laser points surrounded by circles in Figure 6.23. The highlighted laser points introduce ambiguity in the clustering process.

In the second scenario (the vehicle partially occluded by another), the same remark can be formulated considering the left part of the considered zone. There are, however, some errors appearing near to the registration plate. In the right part, a high gap of disparity values is visible. Again, this area corresponds to the ROI defined by the two projected laser points surrounded by circles in Figure 6.23.

Figure 6.25 illustrates the zoom of the disparity maps showing the ROIs

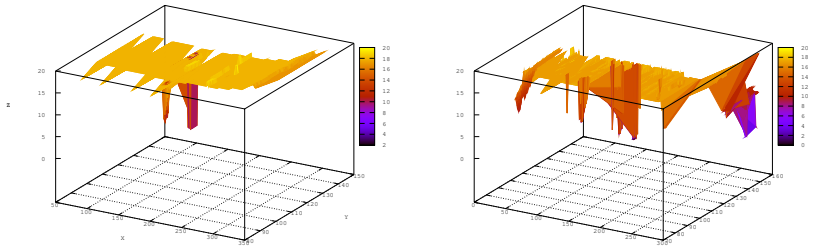

Figure 6.24: Surface maps: the "vehicle turning" situation (left); the "vehicle occlusion" situation (right).

with their neighborhoods.

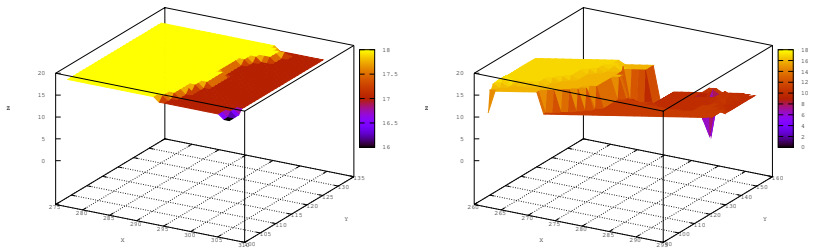

Figure 6.25: Zoom of the surface maps around the ROIs.

6.3.4 Discontinuity test

To decide if two consecutive laser points belong to a same object, the disparity map of the region of interest (ROI) defined by the projected laser points is analysed. The disparity map analysis consists of detecting discontinuities between the projected laser points.

To achieve that, a disparity map exploration process is developed in order to check if a discontinuity-less path between the projected laser points exist. If a discontinuity-less path is found, then the two laser points are considered to belong to the same object. Otherwise, the two points belong to different objects. Two map exploration algorithms are proposed.

The first algorithm is based on the assumption that points are connected by planar surface, and is expressed as follows. After projecting the considered two laser points into the images, the disparity values are calculated only for pixels lying on the line segment connecting the two laser points.

In the next step, for each pair of consecutive disparity values, a discontinuity is detected if the absolute value of the difference of the disparity values ($\Delta Disp_{ij}$ for pixels i and j) is greater than the disparity threshold T_{disp}.

The disparity threshold T_{disp} is expressed as follows:

$$T_{disp} = \frac{T_r \times T_d \times \Delta Disp_{0N}}{d} \tag{6.8}$$

where T_r is a weighting coefficient and T_d is a distance threshold. $\Delta Disp_{0N}$ is the difference of the disparity values of the laser points p_0 and p_N, defining the ROI in which the stereovision analysis is performed, and d is the distance between them. The disparity threshold is a real distance threshold expressed in disparity values. The weighting coefficient T_r is introduced to allow taking into account the disparity map imperfections and the fact that disparity difference does not always represent the same real distance for the same stereo image. The real distance d_{ij} computed from the disparity difference $\Delta Disp_{ij}$ is related to the absolute values $Disp(p_i)$ and $Disp(p_j)$ used to obtain the disparity difference. In the presented work, however, the different values of the weighting coefficient are not evaluated and T_r is set to 1.

This algorithm is simple and quick, but is vulnerable to the disparity maps imperfections. In addition, the following assumption must be made: two laser points representing the same object are connected by the line segment lying on a planar surface to which the points belong (see Figure (a)).

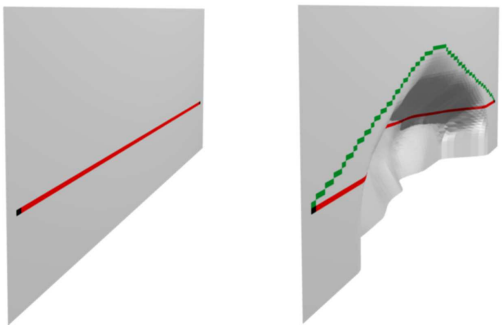

Figure 6.26: Two laser points lying (left) on a planar surface and connected by a line segment (right) on not flat surface and connected by a curve.

The second algorithm evolved from the first one. The disparity values are found for a band of pixels. The band is constructed by expanding in the top and bottom directions the line segment connecting the two laser points (see Figure (b)).

The search is performed according to a disparity rule: two adjacent pixels p_i and p_j belong to the path, if the quantity $\Delta Disp_{ij} = |Disp(p_i) - Disp(p_j)|$ is inferior to a disparity threshold T_{disp}. $Disp(p_i)$ is the p_i pixel disparity value. The disparity threshold T_{disp} is calculated according to the equation (6.8).

Since the disparity maps have always similar topological structure (see Figure 6.25), to optimise the disparity map exploration, a specific algorithm is considered. The algorithm tries to find the path between the two laser points. It must not be the shortest path. The disparity map is considered as a grid graph, where pixels represent the nodes. Each node is connected by edge with its eight neighbours. A value is attributed to each edge. The value, expresses if the edge can be traversed. The values are calculated ad hoc, during the execution of the path search algorithm, and are computed as follows :

$$EV(i,j) = \begin{cases} 0, & \text{if } \Delta Disp_{ij} < T_{disp} \quad (6.9a) \\ \text{inf}, & \text{if } \Delta Disp_{ij} \leq T_{disp} \quad (6.9b) \end{cases}$$

Assigning the value 0 to the edge (i, j) means that the edge can be traversed. Otherwise, the edge in not traversable. The value inf can be interpreted as an obstacle. Each node in the graph has x (i_x) and y (i_y) coordinates.

The searching algorithm is an A* pathfinding algorithm [34] adapted to the treated problem. The algorithm finds a path from a starting node to a goal node. It must not be the shortest path, and thus, there are less constraints. The algorithm is faster and less memory demanding compared with the shortest path A* algorithm.

The initial assumption is that the starting node is situated on the left border of the graph and the goal node is situated on the right border of the graph. There are three different graph exploration modes. The first one is to take the goal node direction for a movement. The movement direction is found by the slope of the line connecting the current and the goal nodes. The algorithm starts with the goal node exploration direction. If a non-walkable edge is found, the algorithm changes to top or bottom exploration modes.

The heuristic function $H(n)$ for a node n is a diagonal distance, defined as:

$$H(n) = max(|i_x - goal_x|, |i_y - goal_y|) \quad (6.10)$$

The path distance $F(n)$ for the node n is always equal 0 since the cost of each traversable edge is 0. Thus, $F(n) = H(n)$. Each open list of nodes is stored with the last chosen directory variable n_{ld}, and expansion variable n_{exp}. The

first variable stores the direction by which the current node is reached. The second one expresses the direction of the expansion, and can takes three values -1, 0 and 1, which represent respectively the top direction expansion, the goal direction expansion, and the bottom direction expansion. The movement direction $dir \in [0, 1, 2...7]$. Due to the assumption of the starting and goal nodes positions, the directions 0,1 and 7 are considered as goal exploration directions.

The algorithm description is as follows:

1. Add the starting node to the open list with the last directory variable $ld = 0$ and expansion variable $exp = 0$.

2. Repeat the following:

   A Look for the lowest H cost node i in the open list. The edges values are equal 0, thus F(i) = H(i). This node will be referred as the current node.

   - If there are nodes with the same H cost
     Than take the one with the greater y (i_y) coordinate.
   - If the selected node has a top expansion direction $i_{exp} = 1$ (respectively a bottom expansion direction $i_{exp} = -1$) and has reached the top border of the graph (respectively the bottom border of the graph)
     Than remove it from the open list, and take the other node with the lowest H cost from the open list.
   - If the selected node is a starting node with an expansion direction $i_{exp} \neq 0$
     Than remove it from the open list and take the other node with lowest H cost from the open list.

   B Switch the current node i to the closed list / Remove the current node i from the open list.

   C Find the node j from the eight nodes, which are adjacent to the current node, by moving in the direction j_{dir}, indicated by the $F_{dir}(i)$ function.
   The $F_{dir}(i)$ function is expressed as follows:

$$F_{dir}(i) = \begin{cases} 1, & \text{if } atan(a) \geq 22.5^\circ \quad (6.11a) \\ 0, & \text{if } -22.5^\circ > atan(a) > 22.5^\circ \quad (6.11b) \\ 7, & \text{if } atan(a) \leq 22.5^\circ \quad (6.11c) \end{cases}$$

where a is the slope of the line connecting the node i and the goal node.

- If the edge $e(i,j)$ is walkable
  Then add the node j to the open list with $j_{ld} = j_{dir}$ and $j_{exp} = 0$.
- Else
  - Find the two nodes j^T, j^B from the eight nodes, which are adjacent to the current node, by moving in the direction j_{dir}, indicated by the $F_{expansionDir}(i^T)$,$F_{expansionDir}(i^B)$. Where the node i^T is a copy of the node i with $i^T_{exp} = 1$ and $i^B_{exp} = -1$. The variables, of the node j are $j^T_{exp} = i^T_{exp}$, $j^T_{ld} = j^T_{dir}$, $j^B_{exp} = i^B_{exp}$, $j^B_{ld} = j^B_{dir}$.
  - Add each node to the open list when $j_{dir} \neq (i_{ld}+4)\ mod\ 8$

D Stop when:

- the goal node is added to the open list. In this case, the path is found
- the open list is empty. In this case, there is no path

The $F_{expansionDir}(i)$ function returns the first walkable direction from the list L_{dir}. L_{dir} is constructed as follows:

- If $i_{ld} \in (3,5)$
  $(i_{ld} + (2+k) * i_{exp})\ mod\ 8$
- Else
  $(i_{ld} + (1+k) * i_{exp})\ mod\ 8$

where $k = 0, 1, 2, ..7$.

6.3.5 Simulator based results

To evaluate the proposed approach, we use a scenario where the ego vehicle follows another one. During its travel (see Figure 6.29), the preceding vehicle avoids a stationary object. At the beginning of the avoidance maneuver, the clustering ambiguities appear for the two objects seen by the LRF: the preceding vehicle and stationary object.

Figures 6.27 and 6.28 present the results of the threshold based clustering for aforementioned clustering ambiguity situations. The gray rectangle corresponds to the real objects. The gray points represent the LRF raw

data points. The red rectangle represents the raw data points cluster. In the case of no-fusion clustering, it is difficult to find a unique threshold allowing to achieve a correct clustering for all the situations. Indeed, the threshold is to small in the case shown in the Figure 6.27(a) and is to big in the case shown in the Figure 6.28(b).

One can see, in Figure 6.28, that the fusion based clustering algorithm produces correct clusters of all the situations.

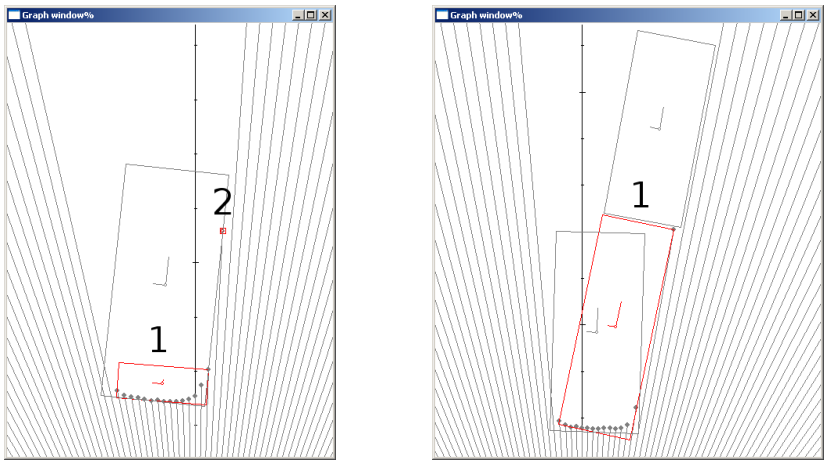

Figure 6.27: Data association results of LRF based clustering for the "vehicle turning" situation (left); and for the "vehicle occlusion" one (right).

Figure 6.28: Data association results LRF-stereovision fusion based data clustering for the "vehicle turning" situation (left); and for the "vehicle occlusion" one (right).

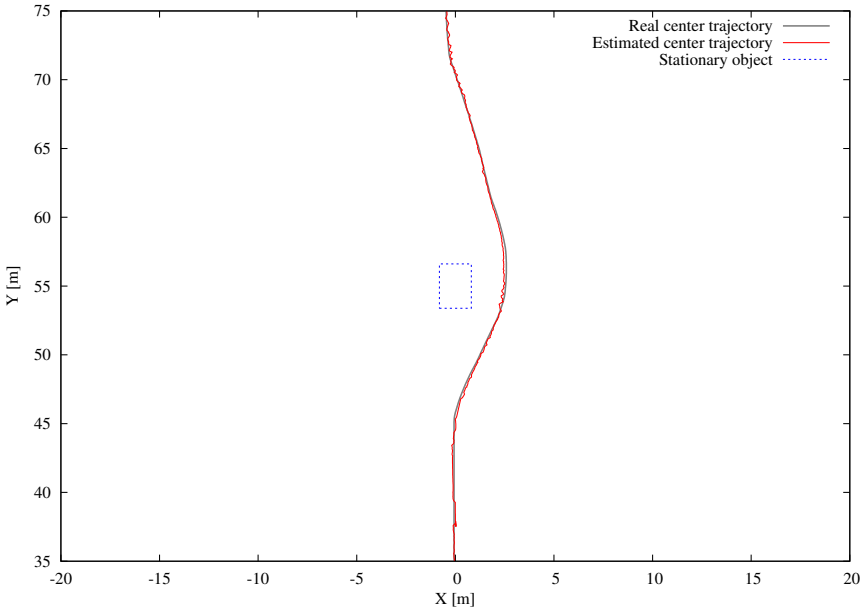

Figure 6.29: Trajectory of the tracked object in absolute coordinate system - with LRF-stereovision fusion.

The results of tracking using the proposed clustering fusion method are presented in figures 6.30-6.33. Figures 6.30 and 6.31 show the estimation errors of the tracked object centre's position in the X and Y axis respectively. Figures 6.32 and 6.33 show the estimation errors of the tracked objects size in the X and Y axis respectively.

One can see that X coordinate centre's position and objects size estimations are more precise than those associated with the Y coordinate (see Figures 6.30-6.33). This is due to the fact that, most of the time, only the X-side of the tracked object is seen. At the beginning of the obstacle avoidance maneuver (around 600th time instant), the Y coordinate related state estimation becomes more precise thanks to the appearance of the second side (Y-side) of the tracked object. One can see also that the Y coordinate related state estimation stays almost unchanged in terms of precision, even if again only one side (X-side) of the object is seen. This is due to the FS assumption, which allows to exploit the most precise object's size estimation, memorized during the tracking. The correct object's states estimation and tracking is guaranteed thanks to the correct LRF data clustering by LRF-Stereovision fusion. Indeed, without stereovision information and when clustering ambiguity situations appear, the tracking fails (creation of many false tracks). From trajectory graph (see Figure 6.29), one can conclude that the points of the stationary object were not put in the same cluster as points originating from moving one. From graphs representing

size estimation errors (Figures 6.32 and 6.33) of the moving vehicle, one can see that all points originating from this vehicle are correctly inserted into one cluster. If it was not the case, the size estimation would be faulty. In our case, however, the Y size evolution is correct. The error of the size reduces when new points appear at the moment of the maneuver.

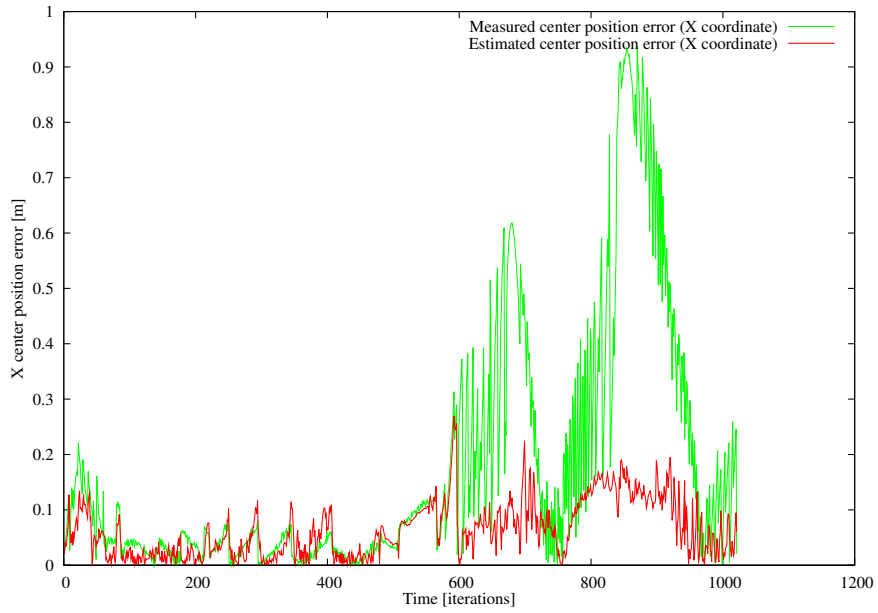

Figure 6.30: Tracked object centre's position error - X coordinate.

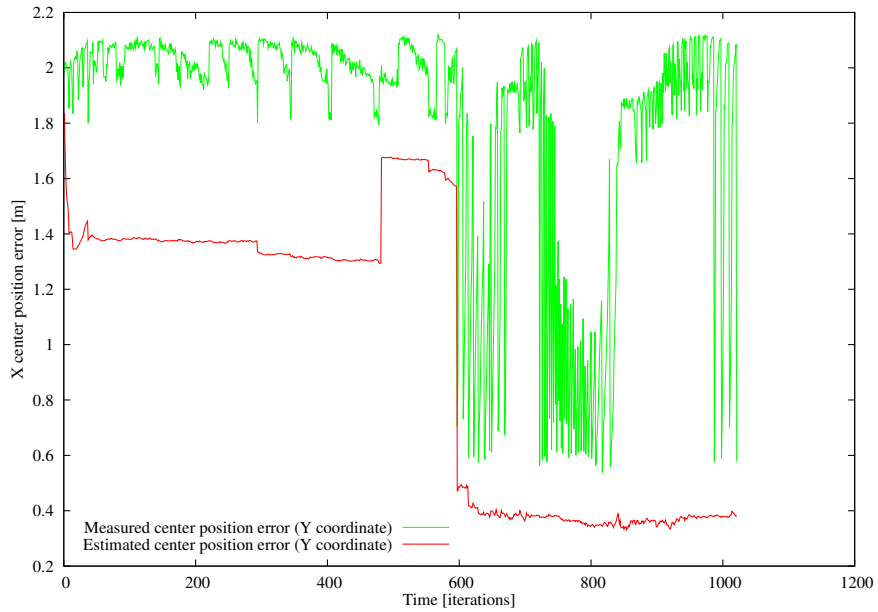

Figure 6.31: Tracked object centre's position error - Y coordinate.

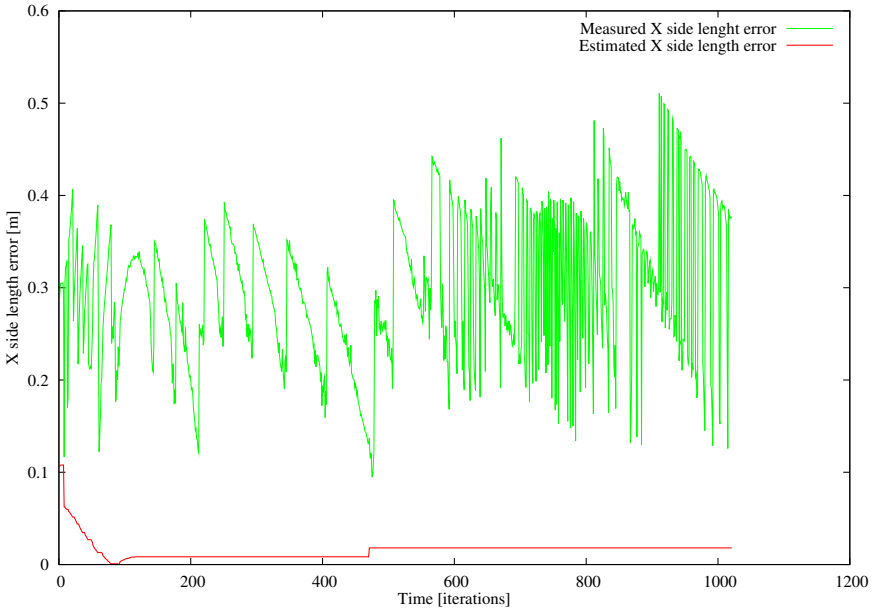

Figure 6.32: Tracked object side error - X coordinate.

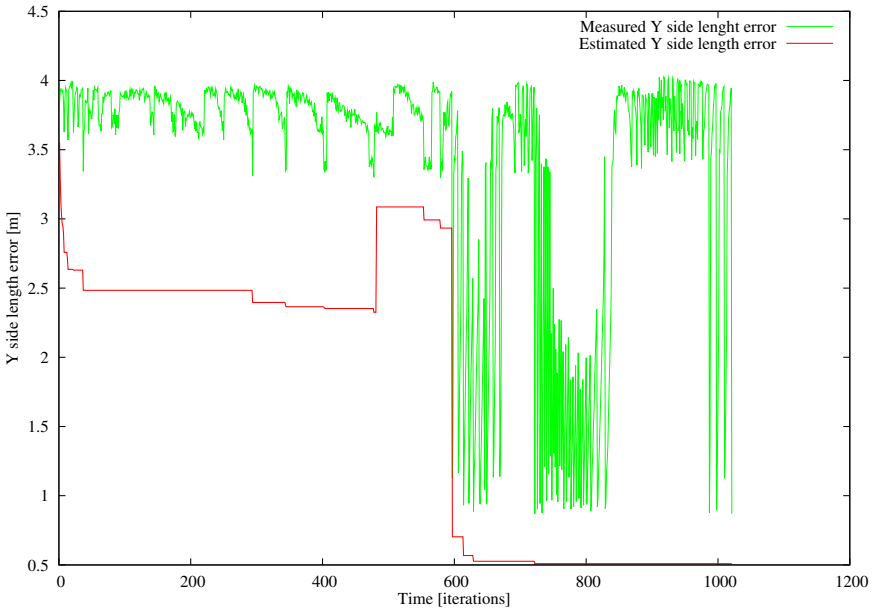

Figure 6.33: Tracked object side error - Y coordinate.

6.3.6 Conclusions

The problem discussed in this section concern clustering of laser data points. A method which integrates stereovision information for extracting laser data points cluster is presented. The fusion algorithm allows removing the clustering ambiguities appearing when only LRF data are used. To avoid a frequent execution of the fusion algorithm, a clustering schema which takes into account that existing tracks is presented. The presented experimental results show the effectiveness and the reliability of the proposed approach.

Chapter 7

Conclusions

The aim of the thesis was to propose algorithms that increase the reliability and robustness of dynamic object tracking. We developed a new object representation model (OBB) relaxing the linear surface assumption, used in a lot of methods. Two paradigms consisting of Inter-Rays (IR) and Fixed Size (FS) assumption are introduced to improve the object state estimates. We proposed a two laser scanner fusion algorithm in order to increase tracking reliability of distant objects. Based on the Nearest-Neighbour principle and the Fixed Size assumption a data association methods is developed to associate laser point with track in case of coalescing objects. Finally a stereovision and laser scanner fusion method is proposed to cluster laser data points.

For in-laboratory experiments, a software simulation platform is developed to test and evaluate proposed algorithms on virtual scenarios. The methods were also evaluated on real conditions using a vehicle research platform.

The OBB representation model and OBB extraction algorithm is based on rotating calipers and on-line convex-hull creation. Using experiments based on the simulator and the real vehicle platform, it was shown by simulator and real vehicle platform base experiments that the proposed object representation model is adequate to tracking dynamic objects. There are, however, some points configuration where the orientation angle of the extracted OBB is not optimal. In the perspective, we propose to use an angle gating based approach to increase the reliability of rotating callipers based algorithm. The angle gating could be based on the velocity prediction. For the angle gating the fact that OBB cannot intersect other lasers ray than

the ones which produced laser points for which OBB was extracted. In the second angle gating approach, the size estimation obtained by the Fixed Size principle will be a crucial part. Side visibility detection is another aspect of the OBB extraction which should be considered. In the perspective, we think about a method which will provide information about the visibility of the object sides basing on the extracted OBB and data points. The proposed visibility detection approach is based on the orientation relation between the extracted OBB and the laser scanner sensor.

To increase the reality of the size and position estimation the Inter-Rays (IR) uncertainty notion was introduced. This algorithm is based on the fact that the raw data points representing the extremities of the extracted OBB represent rarely the real object's extremities. The IR uncertainty combined with the Fixed Size assumption, allows to increase the accuracy of the size and position estimation. The Fixed Size (FS) assumption is based on the observation, that, in general tracked objects do not change their size during the tracking. The algorithm stores the best size estimation, in terms of uncertainty, obtained up to the current moment. To achieve this, it stores the maximal perceived size and the minimal IR uncertainty obtained in the previous measurements. The characteristics of this approach were evaluated using both simulated and real data. From a theoretical point of view, the proposed method is not optimal. Thus, the perspective is to develop a method which will calculate the size estimation and uncertainty from theoretical maximal size and maximal perceived size obtained during the tracking.

The tracking was based on Extended Kalman Filter with Discrete White Acceleration Noise kinematic model. It will be interesting to test and compare other algorithms like Discrete Wienner Process Noise kinematic model, Unscented Kalman Filter, interacting multiple model, Particle Filter.

The method of two laser scanner fusion was proposed to increase the tracking accuracy in terms of object size, orientation angle and velocity estimation. The method is dedicated to be used with presented earlier OBB extraction method. Simulator based evaluation showed that the two laser scanner configuration increase the perception angular resolution of the system, and thus, allows to obtain more precise estimation of object size, orientation angle and velocity. It was shown that this approach increases tracking robustness, especially for distant objects, when compared to the approach using a single laser scanner.

The aspect of data association was also treated in this thesis. The general data association approach was presented, followed by the description

of two methods. Based on the Nearest-Neighbourhood (NN) principle and the FS assumption, the first one was proposed for associating the laser data points with tracks in case of coalescing objects. Combination of the NN approach with additional size information obtained by FS algorithm results in quick and effective algorithm adapted to OBB based object representation. Based on simulated and real data it was shown that the method gives reliable results even in the cases where objects touch each other. One of the perspectives is to develop a probability based association method, adapted to the OBB representation model, for less regular objects like pedestrians.

The second data association method is based on stereovision and laser scanner. We were interested on the clustering problem, which is generally the first stage in the data association process. A fusion method, which uses depth information for laser data points clustering was presented. The additional stereovision information brings more insight into the observed scene, and thus, it is possible to cluster correctly ambiguous laser data points configurations. The method does not assume that objects are represented by planar surfaces and does not assume that only consecutive points can originate from the same surface. Nevertheless, the performance of the method depends on the quality of the obtained disparity map and on the images resolution. Thus, it is planed to perform a more reliable disparity map generation algorithm, including a method which uses laser scanner data for reducing the search space during the matching process.

Another problem, emerged during the experiments, is the detection of low reflective objects (eg. black cars). A three sensor configuration including three sensors: milimeterwave radar, stereovision and laser scanner is in consideration.

The presented works assume that the tracking is performed in flat environment. It is planned to remove this assumption by integrating the estimation of the road surface using stereovision and laser scanner data.

Bibliography

[1] http://www.robosoft.com/eng/. [cited at p. 48]

[2] M.D. Adams. On-line gradient based surface discontinuity detection for outdoor scanning range sensors. In *Proc. of Int. Conf. on Intelligent Robots and Systems*, volume 3, pages 1726 – 1731. IEEE/RSJ, 2001. [cited at p. 10, 11]

[3] K. O. Arras. An introduction to error propagation: Derivation, meaning and examples cy = fx cx fx'. Technical Report EPFL-ASL-TR-98-01 R3, Autonomous Systems Lab, Institute of Robotic Systems, Swiss Federal Institute of Technology Lausanne, 1998. [cited at p. 60]

[4] K. O. Arras. *Feature-Based Robot Navigation in Known and Unknown Environments*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 2003. [cited at p. 14]

[5] K.O. Arras and R.Y. Siegwart. Feature extraction and scene interpretation for map-based navigation and map building. In *SPIE, Mobile Robotics XII*, volume 3210, pages 42–53, Pittsburgh PA, USA, October 1997. [cited at p. 12, 13, 14, 56, 61]

[6] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2001. [cited at p. 24]

[7] H. Baltzakis, A. Argyros, and P. Trahanias. Fusion of laser and visual data for robot motion planning and collision avoidance. *Machine Vision and Applications*, 15:92–100, 2002. [cited at p. 36]

[8] Y. Bar-Shalom and L. Campo. The effect of the common process noise on the two-sensor fused-track covariance. *Transactions on Aerospace and Electronic Systems*, 22(6):803–805, 1986. [cited at p. 33]

[9] Y. Bar-Shalom and T.E. Fortman. *Tracking and data association*. Academic Press Professional, Inc., 1988. [cited at p. 11, 25, 27, 33]

[10] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. *Estimation with Applications To Tracking and Navigation*. Wiley-Interscience, 2001. [cited at p. 11, 67]

[11] Y. Bar-Shalom and X.R. Li. *Multi-target-Multisensor Tracking: Principles and Techniques*. Storrs, CT: YBS Publishing, 1995. [cited at p. 26, 27]

[12] S. Barnard and M. Fisher. Computational stereo. *ACM Computational Surveys*, 14:553–572, 1982. [cited at p. 40]

[13] S.T. Barnard and W.B. Thompson. Disparity analysis of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4):333–340, July 1980. [cited at p. 40]

[14] S. Blackman. *Multiple-Target Tracking with Radar Applications*. Artech House, 1986. [cited at p. 25]

[15] C. Blanc, L. Trassoudaine, and J. Gallice. Ekf and particle filter track-to-track fusion: a quantitative comparison from radar/lidar obstacle tracks. In *Proc. of Int. Conf. on Information Fusion*, volume 2, pages 1303–1310, 2005. [cited at p. 11]

[16] C. Blanc, L. Trassoudaine, Y.L. Guilloux, and R. Moreira. Track to track fusion method applied to road obstacle detection. In *Proc. of Int. Conf. on Information Fusion*, volume 2, pages 775–782, 2004. [cited at p. 11, 33]

[17] G.A. Borges and M.J. Aldon. Line extraction in 2d range images for mobile robotics. *Journal of Intelligent and Robotic Systems*, 40(3):267–297, 2004. [cited at p. 8, 9, 10, 128]

[18] P.L.M. Bouttefroy, A. Bouzerdoum, S.L. Phung, and A. Beghdadi. Vehicle tracking using projective particle filter. In *Proc. of Int. Conf. on Advanced Video and Signal Based Surveillance*, pages 7–12, Los Alamitos, CA, USA, 2009. IEEE. [cited at p. 24]

[19] S. Chambon. *Mise en correspondance stéréoscopique d'images couleur en présence d'occultations*. PhD thesis, Université de Paul Sabatier de Toulouse III, 2005. [cited at p. 38, 144, 145]

[20] U.R. Dhond and J.K. Aggarwal. Structure from stereo - a review. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1489–1510, 1989. [cited at p. 40]

[21] K.C.J. Dietmayer, J. Sparbert, and D. Streller. Model based object classification and object tracking in traffic scenes from range images. In *Proc. of Intelligent Vehicles Symposium*, pages 25–30, Tokyo, 2001. IEEE. [cited at p. 7, 8, 128]

[22] L. DiStefano, M. Marchionni, S. Mattocia, and G. Neri. A fast area-based stereo matching algorithm. In *Proc. of Int. Conf. on Vision Interface (VI)*, 2002. [cited at p. 144]

[23] D. Dooze. *Conception et rTalisation dun stTrToscope bimodal a portTe variable: Application a la dTtection dobstacles a lavant de vThicules guidTs automatisTs*. PhD thesis, UniversitT des Sciences et Technologies de Lille, France, 2001. [cited at p. 40]

[24] R. Duda and P. Hart. *Pattern classification and scene analysis*. John Wiley and Sons, Inc, NY, 1973. [cited at p. 14, 36]

[25] T.E. Fortman, Y. Bar-Shalom, and M. Scheffe. Multi-target tracking using joint probabilistic data association. In *Proc. of Decision and Control Conference including the Symposium on Adaptive Processes*, volume 19, pages 807–812, 1980. [cited at p. 11]

[26] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173– 184, july 1983. [cited at p. 11]

[27] H. Freeman and R. Shapira. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Communications of ACM*, 18(7):409–413, July 1975. [cited at p. 16]

[28] K.Ch. Fuerstenberg and M. Dittmer. Data fusion and ego-motion estimation using laserscanners. In *Proc. of Intelligent Transportation Systems*, volume 2, pages 1224– 1228, 2003. [cited at p. 36, 37]

[29] Q. Gan and C. Harris. Comparison of two measurement fusion methods for kalman-filter-based multisensor data fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 37(1):273–280, 2001. [cited at p. 33]

[30] J.B. Gao and C.J. Harris. Some remarks on kalman filters for the multisensor fusion. *Information Fusion*, 3(3):191–201, September 2002. [cited at p. 34, 35, 112]

[31] F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, pages 425–437, 2002. [cited at p. 24]

[32] R.M. Haralick and L.G. Shapiro. *Image matching, Computer and Robot Vision, Part 2*. Addison-Wesley, New York, 1992. [cited at p. 40]

[33] C. Harris, A. Bailey, and T. Dodd. Multisensor data fusion in defense and aerospace. *Journal of Royal Aerospace Soceity*, (162(1015)):229–244, 1998. [cited at p. 32]

[34] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. [cited at p. 149]

[35] H. Hirschmuller. Improvements in real-time correlation-based stereovision. In *Workshop on Stereo and Multi-Baseline Vision*, pages 141–148, 2001. [cited at p. 144]

[36] U. Hofmann, A. Rieder, and D. Dickmanns. Radar and vision data fusion for hybrid adaptive cruise control on highways. *Machine Vision and Applications*, 14:42–49, 2003. [cited at p. 36]

[37] P.V.C. Hough. Method and means for recognising complex patterns. *US Patent 3069654*. [cited at p. 14]

[38] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 30(3):264–323, 1999. [cited at p. 5]

[39] B. Jhne and H. Haussecker. *Computer vision and applications*. Academic Press, 2000. [cited at p. 40]

[40] B. Jida. *Intégration du Contexte par Réseaux Bayesien pour la Détection et le Suivi Multi-Cibles*. PhD thesis, Universite du Littoral Cote D'Opale, 2008. [cited at p. 6, 8]

[41] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proc. of Int. Symposium on Aerospace/Defense Sensing, Simulation and Controls, Orlando, Floride, USA*, 1997. [cited at p. 21]

[42] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, 2000. [cited at p. 21]

[43] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the AMSE - Journal of Basic Engineering*, (82):34–45, 1960. [cited at p. 18]

[44] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Actvie contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988. [cited at p. 40]

[45] C. Kim, K.-M. Lee, B.-T. Choi, and S.-U. Lee. A dense stereo matching using two-pass dynamic programming with generalized ground control points. In *Proc. of Conf. of Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1075–1082. IEEE, 2005. [cited at p. 144]

[46] P. Kmiotek and Y. Ruichek. Multisensor fusion based tracking of coalescing objects in urban environment for an autonomous vehicle navigation. In *Proc. of Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pages 52–57, Seoul, 2008. IEEE. [cited at p. 112]

[47] A. N. Kolmogorov. *Interpolation and extrapolation of stationary random sequences*, volume 5. 1941. [cited at p. 17]

[48] R. Labayrade, D. Aubert, and J.-P. Tarel. Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation. In *Proc. of Intelligent Vehicle Symposium*, pages 646–651. IEEE, 2002. [cited at p. 37]

[49] K.J. Lee. Reactive navigation for an outdoor autonomous vehicle. Master's thesis, University of Sydney, Department of Mechanical and Mechatronic Engineering, 2001. [cited at p. 7, 128]

[50] D.G. Lowe. Object recognition from local scale-invariant features. In *Proc. of Int. Conf. on Computer Vision*, volume 2, page 11501157, 1999. [cited at p. 40]

[51] R.C. Luo, C.-C. Yih, and K.L. Su. Multisensor fusion and integration: Approaches, applications, and future research directions. *IEEE Sensors Journal*, 2(2):107–119, April 2002. [cited at p. 31]

[52] D. Marr and T. Poggio. A computational theory of human stereovision. In *Proc. of the Royal Society of London*, volume 204, issue 1156, pages 301–328, 1979. [cited at p. 40]

[53] T. Marr, D. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283287, 1976. [cited at p. 40]

[54] Ch. Mertz, D. Duggins, J. Gowdy, J. Kozar, R. MacLachlan, A. Steinfeld, A. Suppe, Ch. Thorpe, and Ch. Wang. Collision warning and sensor data processing in urban areas. In *Proc. of Int. Conf. on ITS telecommunications*, pages 73–78, Brest, 2005. [cited at p. 7]

[55] H. Moravec. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. PhD thesis, Computer Science Department, Stanford University, March 1980. 433. [cited at p. 40]

[56] V. Nguyen, A. Martinelli, N. Tomatis, and R.Y. Siegwart. A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics. In *Proc. of Int. Conf. on Intelligent RObots and Systems*, pages 1929– 1934, Edmonton, Canada, 2005. IEEE/RSJ. [cited at p. 14, 55]

[57] T. Pavlidis and S.L. Horowitz. Segmentation of plane curves. *IEEE Transactions on computers*, c-23(8):860–870, 1974. [cited at p. 14, 55]

[58] M. Perrollaz, R. Labayrade, C. Royere, N. Hautiere, and D. Aubert. Long range obstacle detection using laser scanner and stereovision. In *Proc. of Intelligent Vehicle Symposium*, pages 182–187, Tokyo, 2006. [cited at p. 37]

[59] A. Petrovskaya and S. Thrun. Model based vehicle tracking for autonomous driving in urban environments. In *Proc. of Robotics: Science and Systems Conference*. RSS Robotics, 2008. [cited at p. 7, 15, 16]

[60] C. Premebida and U. Nunes. Segmentation and geometric primitives extraction from 2d laser range data for mobile robot applications. In *Proc. of Nat. Festival of Robotics Scientific Meeting (ROBOTICA)*, pages 17–25, Coimbra, 2005. [cited at p. 6]

[61] Y. Rathi, N. Vaswani, and A. Tannenbaum. A generic framework for tracking using particle filter with dynamic shape prior. *IP*, 16(5):1370–1382, May 2007. [cited at p. 24]

[62] D.B. Reid. An algorithm for tracking multiple targets. *Transaction on Automatic Control*, 24(6):843–854, 1979. [cited at p. 27, 28]

[63] J. Roecker and C. McGillem. Comparison of two-sensor tracking methods based on state vector fusion and measurement fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 24(4):447–449, 1988. [cited at p. 33]

[64] Y. Ruichek and J.G. Postaire. A neural matching algorithm for 3d reconstruction from stereo pairs of linear images. *Pattern Recognition Letters*, 17:387398, 1996. [cited at p. 40]

[65] D. Santos, J.E. Faria, F. Soares, R. Araujo, and U. Nunes. Tracking of multi-obstacles with laser range data for autonomous vehicles. In *Proc. of Nat. Festival of Robotics Scientific Meeting (ROBOTICA)*, pages 59–65, Lisbon, 2003. IEEE. [cited at p. 7, 8, 16, 128]

[66] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002. [cited at p. 40]

[67] G. A. Bekey S.I. Roumeliotis. Segments a layered dual-kalman filter algorithm for indoor feature extraction. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 454–461. IEEE/RSJ, 2000. [cited at p. 10]

[68] A. Siadat, A. Kaske, S. Klausman, M. Dufaut, and R. Husson. An optimized segmentation method for a 2d laser-scanner applied to mobile robot navigation. In *Proc. of Symposium on Intelligent Components and Instruments for Control Applications*, pages 153–158, 1997. [cited at p. 12, 13, 55]

[69] Q. Song and M. Shepperd. Missing data imputation techniques. *Int. Journal of Business Intelligence and Data Mining*, 2(3):261–291, 2007. [cited at p. 24]

[70] S. Streller, K.C.J. Dietmayer, and J. Sparbert. Object tracking in traffic scenes with multi-hypothesis approach using laser range images. In *Proc. of Int. Conf. on Intelligent Transportation Systems(ITSC)*. IEEE, 2001. [cited at p. 15, 16]

[71] L. Tang, C. Wu, and Z. Chen. Image dense matching based on region growth with adaptive window. *Pattern Recognition Letters*, 23:1169–1178, 2002. [cited at p. 40]

[72] R.M. Taylor and P.J. Probertt. Range finding and feature extraction by segmentation of images for mobile robot navigation. In *Proc. of Int. Conf. on Robotics and Automation*, pages 95–100, Minneapolis, Minnesota, 1996. [cited at p. 14]

[73] G. Toussaint. Solving geometric problems with the rotating calipers. In *Proc. MELECON, Athens, Greece*, 1983. [cited at p. 53, 58]

[74] Carnegie Mellon University Robotics Institute University of California PATH. Integrated collision warning system final technical report. Technical report, 2004. [cited at p. 14, 16]

[75] J. Vandorpe, H.V. Brussel, and H. Xu. Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2d range finder. In *Proc. of Int. Conf. on Robotics and Automation(ICRA)*, pages 901–908, Minneapolis, Minnesota, April 1996. IEEE. [cited at p. 11, 13]

[76] E. Wan and R. Van der Merwe. The unscented kalman filter for nonlinear estimation. *IEEE Symposium Adaptive Systems for Signal Processing, Communication and Control (AS-SPCC 2000), Lake Louise, Alberta, Canada*, pages 153–158, October 2000. [cited at p. 21]

[77] J.H. Wang and C.P. Hsiao. On disparity matching in stereo vision via a neural network framework. In *Proc. Natl. Sci. Counc. ROC(A),*, volume 23, pages 665–678, 1999. [cited at p. 40]

[78] N. Wiener. *Extrapolation, Interpolation and Smoothing of Time Series, with Engineering Applications*. New York Wiley, 1949. [cited at p. 17]

[79] N. Wiener and E. Hopf. On a class of singular integral equations. In *In Proc. Prussian Acad. Math. Phys. Ser.*, page 696, 1931. [cited at p. 17]

[80] D. Willner, C.B. Chang, and K.P. Dunn. Kalman filter algorithms for a multi-sensor system. In *Proc. of Conf. on Decision and Control including Symposium on Adaptive Processes*, volume 15, pages 570–574. IEEE, 1976. [cited at p. 33]

Publications :

1. Pawel Kmiotek, Yassine Ruichek, Multisensor fusion based tracking of coalescing objects in urban environment for an autonomous vehicle navigation, IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008), Korea University Seoul Korea, August 20-22 2008

2. Pawel Kmiotek, Yassine Ruichek, An Oriented Bounding Box Representation Method for Tracking Dynamic Objects, Joint 4th International Conference on Soft Computing and Intelligent Systems and 9th International Symposium on advanced Intelligent Systems (SCIS & ISIS 2008), Nagoya Japan, September 17-21 2008

3. Pawel Kmiotek, Yassine Ruichek, Objects Oriented Bounding Box based Representation using Laser Range Finder Sensory Data, IEEE International Conference on Vehicular Electronics and Safety (ICVES 2008), Columbus Ohio USA, September 22-24 2008

4. Pawel Kmiotek, Yassine Ruichek, Representing and Tracking of Dynamics Objects using Oriented Bounding Box and Extended Kalman Filter, IEEE Conference on Intelligent Transportation Systems (ITSC 2008), Beijing China, October 12-15, 2008

5. Pawel Kmiotek, Yassine Ruichek, Two laser scanners raw sensory data fusion for objects tracking using Inter-Rays uncertainty and a Fixed Size assumption., The 12th International Conference on Information Fusion (FUSION 2009), Seattle USA, July 6-9, 2009

6. Pawel Kmiotek, Cyril Meurie, Frederick Zann, Yassine Ruichek, A clustering of range sensory data by integrating stereovision information for tracking objects in urban environment., Models and Technologies for Intelligent Transportation Systems (MTITS 2009), Rome Itally, June 22-23, 2009

7. Pawel Kmiotek, Yassine Ruichek, "Inter-Rays uncertainty and Fixed Size assumption for objects tracking using a Laser Scanner ", The 2nd IFAC International Conference on Intelligent Control Systems and Signal Processing (ICONS 2009), Istanbul Turkey, September 21-23, 2009

8. Pawel Kmiotek, Cyril Meurie, Yassine Ruichek, Frederick Zann, "A LRF and stereovision based data association method for objects tracking", IEEE International Conference on Systems, Man, and Cybernetics (SMC 2009), San Antonio USA, October 11-14, 2009

9. Pawel Kmiotek, Yassine Ruichek, "A Fixed Size Assumption Based Data Association Method for Coalescing Objects Tracking using a Laser Scanner Sensor", The 2009 IEEE International Conference On Vehicular Electronics and Safety (ICVES 2009), Pune India November 10-12, 2009