DYNAMIC OBJECT DETECTION AND TRACKING USING SENSOR FUSION AND PARTICLE FILTER

(SENSÖR FÜZYONU VE PARÇACIK FİLTRELEMESİ KULLANARAK HAREKETLİ NESNELERİN TESPİTİ VE TAKİP EDİLMESİ)

by

Berk PELENK, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Date of Submission : May 22, 2013

Date of Defense Examination : June 13, 2013

Supervisor : Assoc. Prof. Dr. Tankut ACARMAN

Committee Members : Assoc. Prof. Dr. S. Emre ALPTEKİN

Asst. Prof. Dr. Murat AKIN

Acknowledgements

My deepest gratitude goes to my supervisor, Assoc. Prof. Dr. Tankut ACARMAN who assisted and inspired me from the beginning to the end of my dissertation, giving me continuous guidance. Moreover, his effective communication skills, patient attitude and quick feedback gave a sense of direction to my study, rendering its completion possible.

I would also like to thank all those people who took time out of their busy lives to support me during my project.

Finally, special thanks goes to my family for their invaluable support during my studies.

Berk PELENK

May, 2013

Table of Contents

Acknowledgements....ii
Table of Contents....iii
List of Symbols / Abbreviations....v
List of Figures....vi
List of Tables....viii
Abstract....ix
Résumé....x
Özet....xi
1 . Introduction....1
2 . System Design and Approach....6
2.1 . System Design....6
2.1.1 . SICK LIDAR LMS – 500....7
2.1.1.1 . What is LIDAR?....7
2.1.1.2 . General/Technical Description....7
2.1.1.3 . LIDAR Operating Principles....10
2.1.2 . Crossbow Technology IMU440CA-200....17
2.1.2.1 . What is IMU (Inertial Measurement Unit) ?....17
2.1.2.2 . General/Technical Specifications....18
2.1.3 . Roboteq AX 1500 Motor Controller & Odometer....21
2.1.4 . SiRFstarIII GPS Receiver....24
2.1.4.1 . What is GPS?....24
2.1.4.2 . General/Technical Details....26
2.1.5 . Entire System Overview....28
2.2 . Main Steps....30
3 . Fusion and Detection Methodology....33
3.1 . General Concept....33
3.2 . Getting LIDAR Returns....33
3.3 . Detecting Clusters from LIDAR Returns....34

3.4 . Normalizing Existing Clusters with IMU and Odometer Data.....36
3.5 . Implementation of Particle Filter.....36
3.6 . Comparison and Identification of Clusters.....39
4 . Experimental Results.....40
4.1 . Test Conditions, Metrics and Summary.....40
4.2 . Tests.....44
4.2.1 . Range Test.....44
4.2.2 . Time Test.....44
4.2.3 . Test with Fixed LIDAR Position + Fixed Clusters.....45
4.2.4 . Test with Fixed LIDAR Position + Dynamic Clusters (Human).....49
4.2.5 . Test with Fixed LIDAR Position + Dynamic Clusters (Car).....53
4.2.6 . Second Test with Fixed LIDAR Position + Dynamic Clusters (Car).....57
4.2.7 . Test with IMU, Odometer, GPS and Particle Filter.....65
5 . Application.....67
5.1 . Overview.....67
5.2 . Detailed Description.....67
5.2.1 . LIDAR Communication.....67
5.2.2 . IMU Communication.....68
5.2.3 . Odometer Communication.....69
5.2.4 . GPS Communication.....70
6 . Performance Analysis.....71
6.1 . CPU Sampling.....71
6.2 . Tracking Memory Allocation.....73
6.3 . Concurrency.....73
7 . Conclusion.....76
References.....77
Biographical Sketch.....81

List of Symbols / Abbreviations

GPS	Global Positioning System
IMU	Inertial Measurement Unit
LIDAR	Light Detection and Ranging
$X_{est}(t)$	Estimated X position in time T from Particle Filter
$Y_{est}(t)$	Estimated Y position in time T from Particle Filter
$\sum x_i$	Sum of x variables
N_{eff}	Particle Filter calculation accuracy

List of Figures

Fig. 2.1. LIDAR LMS-500....8
Fig. 2.2. Measuring principle of the LIDAR....11
Fig. 2.3. Principle of operation for pulse propagation time measurement....12
Fig. 2.4. Reflection of the laser beam at the surface of an object....13
Fig. 2.5. Reflection angle....14
Fig. 2.6. Degree of reflection....14
Fig. 2.7. Mirror surfaces....15
Fig. 2.8. Object smaller than diameter of the laser beam....15
Fig. 2.9. Scanning range of the LIDAR as a function of the target remission....16
Fig. 2.10. IMU accelerometer and angular axis directions....18
Fig. 2.11. Crossbow IMU440CA-200....18
Fig. 2.12. Roboteq AX1500 mainboard....21
Fig. 2.13. SiRFstarIII GPS Receiver 26
Fig. 2.14. A screen-shot of system design from front view....28
Fig. 2.15. A screen-shot of system design from back view....29
Fig. 2.16. A screen-shot of IMU....29
Fig. 2.17. Entire system data flow....32
Fig. 3.1. Shows how the calculation is done for minimum possible distance between two consecutive points of LIDAR data....35
Fig. 4.1. Shows a screen-shot from centroid changes of 4 different fixed objects after 70 degree clockwise rotation of the Lidar. IMU yaw rate data is used to rotate the existing cluster centroids....40
Fig. 4.2. A camera view of a dynamic environment which will be analyzed with LIDAR and clustered....46
Fig. 4.3. Non-clustered view of Figure 4.2....46
Fig. 4.4. Clustered view of Figure 4.2....47
Fig. 4.5. A camera view of the second test from a dynamic environment....48
Fig. 4.6. Non-clustered view of Figure 4.5....48

Fig. 4.7. Clustered view of Figure 4.5.....49
Fig. 4.8. Clustered view of a moving human in a dynamic environment.....50
Fig. 4.9. Clustered view of a moving human after 2 seconds from screenshot time of Figure 4.8.....51
Fig. 4.10. Clustered view of two moving humans.....52
Fig. 4.11. Clustered view of two moving humans after 2 seconds from screenshot time of Figure 4.10.....52
Fig. 4.12. Shows detection and tracking of test scene 1.....54
Fig. 4.13. Shows detection and tracking of test scene 2.....55
Fig. 4.14. Shows detection and tracking of test scene 3.....56
Fig. 4.15. Shows detection and tracking of test scene 4.....57
Fig. 4.16. Shows detection and tracking of test 2 scene 1.....58
Fig. 4.17. Shows occlusion between cars with ID 29 and 11 of test 2 scene 2.....59
Fig. 4.18. Shows occlusion between cars with ID 29 and 11 of test 2 scene 3.....60
Fig. 4.19. Shows occlusion between cars with ID 29 and 11 of test 2 scene 4.....61
Fig. 4.20. Shows occlusion between cars with ID 29 and 11 of test 2 scene 5.....62
Fig. 4.21. Shows occlusion between cars with ID 29 and 11 of test 2 scene 6.....63
Fig. 4.22. Shows occlusion between cars with ID 29 and 11 of test 2 scene 7.....64
Fig. 4.23. Shows occlusion between cars with ID 29 and 11 of test 2 scene 8.....65
Fig. 4.24. Shows the Particle Filter effect to the velocity of a dynamic LIDAR.....66
Fig. 5.1. Shows the LIDAR control window of the software.....68
Fig. 5.2. Shows the IMU control window of the software.....69
Fig. 5.3. Shows the Odometer control window of the software.....70
Fig. 5.4. Shows the GPS control window of the software.....70
Fig. 6.1. Shows CPU usage of the software without Particle Filter.....72
Fig. 6.2. Shows CPU usage of the software with Particle Filter.....73
Fig. 6.3. Shows per-thread summary.....74
Fig. 6.4. Shows the timeline profile.....75
Fig. 6.5. Shows timeline for different worker threads as well as disk read/write.....75

List of Tables

Table 2.1. LIDAR LMS-500 Technical Details.....8
Table 2.2. Crossbow IMU440CA-200 Technical Details.....19
Table 2.3. Roboteq AX1500 Technical Details.....22
Table 2.4. SiRFstarIII GPS Receiver Technical Details.....27
Table 4.1. Detection accuracy in summary for fixed positions of clusters and LIDAR scanner in 15 seconds interval.....41
Table 4.2. Detection accuracy in summary for fixed positions of clusters and LIDAR scanner in 30 seconds interval.....41
Table 4.3. Detection accuracy in summary for fixed LIDAR and dynamic Clusters in 15 seconds interval.....41
Table 4.4.Detection accuracy in summary for fixed LIDAR and dynamic Clusters in 30 seconds interval.....42
Table 4.5. Detection accuracy in summary for mobile LIDAR and dynamic Clusters in 15 seconds interval.....42
Table 4.6. Detection accuracy in summary for mobile LIDAR and dynamic Clusters in 30 seconds interval.....42
Table 4.7. Time Test.....44
Table 4.8. Time Test.....45

Abstract

Object detection and tracking problem has long been an important topic in the literature. The importance of this subject continues to increase due to the advantages that object detection and tracking provides especially for every day practical life and hence the need for its pervasive use. There are numerous different types of object tracking and various different tracking methods for each type of tracking.

This thesis presents a moving object detection and tracking system with a Particle Filter algorithm. The goal is to build an infrastructure that will allow following an unknown moving object in a region with numerous other dynamic objects.

Several components are used to determine objects; to estimate self-localization; and match the determined objects in the next iteration with the previously determined objects in order to tag each object with a particular identification. Specifically, LIDAR is used to determine the objects, IMU(Inertial Measurement Unit) to estimate relative translation and rotation, Odometer and GPS to help increase the accuracy of the self-position that is calculated by the IMU.

The Particle Filter algorithm predicts self-position, utilizing the data received from both the IMU, the Odometer and the GPS. Computational cost is also taken into account during the clustering and matching stages. Performance and detection accuracy tests are carried out using various sized objects, as well as different environmental settings in order to conduct a comparison analysis for the gathered data.

Résumé

La détection et la poursuite des objets est l'un des problèmes les plus connus dans la littérature depuis longtemps. Surtout, dans la vie quotidienne, grâce aux nombreux avantages, l'utilisation des applications de détection et de poursuite des objets, est devenue une quasi-nécessité aujourd'hui.

Cette thèse se sert de l'algorithme de Filtre de Particules pour la détection et la poursuite des objets dynamiques. Le but essentiel, est de préparer une infrastructure pour la poursuite des objets qui entrent dans la vue de l'observateur et qui sont inconnus avant, spécialement dans les environnements où il y a plusieurs différents types d'objets.

Toujours restant sur la ligne du but essentiel, différents composants ont été pris en compte pour la détection des objets, pour le calcul du déplacement de l'observateur et pour la comparaison des objets détectés en temps t avec les objets détectés en temps t-1 afin de faire l'identification et donc la poursuite. LIDAR est utilisé pour trouver l'angle et la distance des objets par rapport à sa position. IMU (Inertial Measurement Unit) est utilisé pour trouver la translation et la rotation de l'observateur par rapport à la position précédente afin de pouvoir calculer les nouvelles positions des objets qui sont trouvés en temps précédent acceptant qu'ils sont immobiles. Dernièrement, Odomètre et GPS sont utilisés afin d'améliorer la précision de la position calculée de l'observateur et de minimiser la faute de la translation qui sont trouvés par l'aide de IMU.

Les données reçues par IMU, Odomètre et GPS sont passées à l'algorithme de Filtre de Particules, cette transition de données diminue la faute cumulative qui pourrait apparaitre sans cet algorithme. Durant toutes ces procédures, l'utilisation de CPU et temps d'opération total sont tenus en compte pendant les périodes de détection et de l'identification. Il faut aussi clarifier que pour tester le système en entier, plusieurs objets avec des types et des dimensions différents sont utilisés et que tous ces résultats ont été comparés les uns aux autres.

Özet

Nesne tespiti ve takibi literatürde uzun zamandır araştırma konusu olan problemlerdendir. Özellikle gündelik hayatta, nesnelerin tespit ve takip edilmesinin sağladığı birçok avantaj olduğundan, kullanımına çok yaygın şekilde ihtiyaç duyulmakta bu da konunun önemini iyiden iyiye arttırmaktadır. Nesne takibi için çeşitli takip türleri mevcut olup, her takip türüne göre kullanılacak takip yöntemleri farklılık göstermektedir.

Bu tez Parçacık Filtrelemesi Algoritması'ndan yararlanarak hareketli nesnelerin tespit ve takip edilmesini konu alır. Temel amaç, hareketli birçok farklı nesnenin olduğu ortamlarda görüş alanına giren ve daha önceden tanınmayan nesnelerin takip edilebilmesi için bir altyapı hazırlamaktır.

Temel amaç doğrultusunda nesnelerin tespiti, gözlemcinin konum değişiminin hesaplanması ve sonraki adımda tespit edilen nesnelerin bir önceki adımda tespit edilenlerle karşılaştırılarak yapılacak takibi için çeşitli donanımlar kullanılmıştır. Detaylandırmak gerekirse, LIDAR, nesnelerin gözlemciye göre pozisyonunun hesaplanması, dolayısıyla nesnelerin tespiti için kullanılırken, IMU (Inertial Measurement Unit) gözlemcinin bir önceki pozisyona göre olan rotasyon ve translasyonunun hesaplanması amacıyla kullanılır. Son olarak Odometre ve GPS kullanılarak, IMU yardımıyla hesaplanan translasyondaki hatanın minimize edilmesi ve gözlemcinin yer değişiminin daha keskin şekilde hesaplanması hedeflenmiştir.

IMU, Odometre ve GPS'den alınan veriler Parçacık Filtresi Algoritması'na sokulmakta ve bu sayede filtre kullanılmadığı taktirde ortaya çıkacak olan toplamsal hata en aza indirgenmektedir. Tüm bunların yanında nesnelerin tespiti ve eşlenmesi sırasında uygulama tarafından kullanılan işlemci gücü, işlem zamanı gibi veriler de dikkate alınmıştır. Son olarak çeşitli büyüklük ve tipteki nesneler kullanılarak farklı çevrelerde testler yapılmış ve sonuçlar birbirleriyle karşılaştırılmıştır.

1 . Introduction

Object detection and moving object tracking have captured the attention of researchers for many years. Besides its broad application, object detection is often used in autonomous intelligent robots, monitoring and surveillance services, smart rooms, vehicle tracking and biomedical image analysis. Nevertheless, it is difficult to identify/classify objects in environments that are comprised of a large number of moving or colliding agents. It is still a challenging problem to detect and track dynamic objects, in particular for the locations with numerous pedestrians; yet there have been significant improvements.

In the past, many object detection systems were developed. Most of them use digital cameras or imaging techniques with 2D/3D image models for modeling the clusters to increase the detection accuracy. The second approach takes into consideration the use of laser imaging detection which is an optical remote sending technology. It is mainly used to measure the distance to targets by illuminating the target with laser light and then analyzing the reflected light for which the range can vary from 22mm to 305meters. However, the accuracy of the device is negatively correlated with the distance. This thesis is based on the latter approach.

In a more detailed outlook, object tracking has become an important issue for the field of computer vision. With the evolution of computers, many algorithms have been developed to identify objects' movements. As the number of these algorithms increased, navigation systems, tracking systems and surveillance systems have become extremely common. For example, people can be tracked for security purposes, such as at pedestrian crossing zones, using particular object tracking tools and algorithms. Similarly, car tracking systems can be developed to keep a secure distance between two cars with the use of aforementioned techniques.

The current difficulty in this area is the insufficiency of data in tracking particular

objects. This is due to the fact that a single device is not capable of collecting the adequate data required to make accurate detections and approximations. Hence, fusion of measurement devices and combination of their data are required key factors (Durrant-Whyte, 2006). The aim here is to provide better detections, positions, clustering and predictions (Tamas & Lazea, 2010).

Commonly, to detect, identify and track objects, camera and video based systems are used because the shape, color and movement can be really characteristic and can be used to distinguish particular objects. Tracking systems are used for security surveillance systems for banks, stores etc. Using cameras, the whole area can be observed, and after a movement, objects like humans can be distinguished, following the application of several algorithms and functions, and detection of several threats. For example, in (Shan et al., 2007), hand control interface for a robotic wheelchair is developed using several video based devices and algorithms, which can be used to develop people motion detection.

Besides camera-video based systems, laser scanners like LIDAR (Light Detection and Ranging or Laser Imaging Detection and Ranging) can also be used in tracking systems, as in this dissertation. LIDAR can measure the distance to targets by sending laser light to targets by measuring reflected light duration.
In this thesis, 2D-LIDAR is used, but there is also a 3D-LIDAR to detect objects in 3 dimensions. 3D-LIDAR, can also analyze objects by their shape, height and width. Like in (Lee et al., 2010), marine environments are detected and can be tracked using a 3D-LIDAR.

Using LIDAR, Adapted Cruise Control system is implemented into a vehicle using IBEO LUX LIDAR and Monocular Monochrome Camera to observe a vehicle in front of a particular vehicle, and by controlling the brakes, several security reasons can be handled for driver assistance.(Weigel et al., 2009) Overall, there are numerous devices that can be used to accomplish tracking and various examples of object detection systems that can be seen in urban life.

Many studies have been conducted to detect and track objects. Some of these works are based on the LIDAR scanner whereas others are done with a camera. Vision systems are commonly used for object detection, without the use of a LIDAR like in (Franke & Heinrich, 2002). Visual object detection and tracking methods are generally based on a simple segmentation procedure such as the temporal difference of background analysis (Toth & Aach, 2003), while on the contrary, background changes affect these approaches adversely because of the camera motion. There are also some examples of the use of vision systems in combination with LIDAR like in (Neira et al., 1999; Premebida et al., 2007). In (Premebida et al., 2007) Camera and LIDAR are used separately and then the results are combined together with the Bayesian Filter approach.

In detection and tracking, the general focus area is the tracking of pedestrians as in (Broggi et al., 2000; Oren et al., 2000; Sotelo et al. 2006). But vehicle tracking is also one of the popular subjects as in (Fortin et al., 2012). Also, as it is stated in (Premebida et al., 2007), there are several approaches to track vehicles and pedestrians at the same time.

Some approaches like (Gao & Coifman, 2006) only focus on vehicle detection on the road. The method presented in (Gao & Coifman, 2006) is based on the detection of road boundaries. The aim here is to classify cars and road boundaries with the help of image processing techniques without the requirement of a digital map. This approach does not cover any cluster identification, but is used for distinguishing between a non-vehicle cluster and a vehicle-cluster, if they are both present on the road.

There are various different methods that are used for cluster tracking. The most common use is that the calculations are based on center of gravity position of clusters, as in (Yu et al., 2008) and objects are tracked by watching the changes of position of their center of gravity. Differently in (Fortin et al., 2012), instead of the center of gravity approach; clusters are tracked by their line segments and the chosen modeling. The disadvantage of approach is that a model must be chosen and analysed prior to implementing any detection or tracking.

Detection and tracking can be carried out in different circumstances where LIDAR can be mobile or objects can be mobile (Fortin et al., 2012) or LIDAR and objects can be mobile at the same time (Yu et al., 2008). In proposed papers, the approaches generally use a LIDAR on a fixed position while the objects are mobile. Our approach is quite similar with (Yu et al., 2008) in terms of the general concept; but in (Yu et al., 2008), k-nearest neighbors method is used for clustering where we introduce a new clustering algorithm for the clustering stage.

Particle Filter algorithm is commonly used in detection and tracking at different stages. In (Fortin et al., 2012), Particle Filter algorithm is used for the detection of a line segment in scanning laser range data, however, in (Yu et al., 2008), it is used in the tracking stage for attaining the movement condition of the objects.

Yet, it is decided that the main objective of this thesis requires a 2D-LIDAR, an IMU (Inertial Measurement Unit) and an Odometer running within a Particle Filter Algorithm. A 2D-LIDAR is sufficient in detection of moving objects in 3D space with the application of a good clustering and a good prediction of the cluster movement. Thus, the additional video-based system will not be given a role in our implementation.

The main difference between the approach of this thesis and the other approaches in literature is that there is not any calculation with any vision based system in this thesis, however, after examining approaches which does not use any vision based system like in this thesis, it can be clearly said that this thesis approach differ from other approaches mainly on clustering algorithm. Besides, the ability of tracking totally different type of objects without the need of modifying the general algorithm or modelling the shapes of objects to track, is another important difference of this thesis approach. Furthermore, compared to others, a new algorithm is implemented in this thesis approach to handle situations which require dynamic LIDAR positioning. In fact, each of the main differences mentioned in this paragraph can be found in several approaches seperately, but this thesis apprach combines all of these differences. However, because of the difficulty to not only focus on a unique type of object but also on other different objects, this thesis approach can be supported with a vision-based sensor especially to increase

detection and tracking accuracy in situations with changing LIDAR position.
The motivation for considering yet another object detection and tracking method comes from the possibility to extend the current used algorithms with a self-developed algorithm and to build a robot which is controlled by our own implementation. There are many methods for clustering and object tracking, and the difference between each method can be ascribed to the LIDAR-IMU-Odometer data fusion and various clustering algorithm implementations. Also, CPU load caused by each algorithm is distinct and therefore a subject that needs more attention from researchers.

The rest of this thesis is organized as follows. In section II, common usage of object detection and tracking will be analyzed. In section III, related works on object detection and tracking will be briefly described within the literature review. In section II, the principles of the thesis approach will be described and the system design will be reviewed. Section III will explain the implementation of the approach and detailed algorithm including fusion and detection methodology. In section IV, experimental results will be analyzed in detail. Software overview and performance analysis will be presented in sections V and VI respectively. In the final section, conclusions will be drawn and possible directions for future work will be proposed.

2 . System Design and Approach

2.1 . System Design

The main hardware components that will be presented in this section are chosen upon their usability, object detection accuracy and self-position estimation accuracy. Taking into consideration possibility of being in different external environments, movement detection problems and necessity of data fusion from different components, the following components are chosen:

- SICK LIDAR LMS -500
- Crossbow Technology IMU440CA-200
- Roboteq AX 1500 Motor Controller & Odometer
- SiRFstarIII GPS Receiver

Owing to fusion of data coming from both IMU and GPS receiver, it's targeted that LIDAR's self-position accuracy will increase. Not only is LIDAR's self-position estimated with the data coming from IMU and GPS, but also with the odometer data which is another important component mainly used for self-position estimation with the aid of calculated distance from rotation numbers of tires and tire perimeter. By the way, the most important component of all is LIDAR without doubt since the data coming from LIDAR is used as a basis for all components. In fact, the data received from LIDAR is first processed in the clustering stage and then used in fusion with the data received from IMU, GPS and Odometer. Each of the component described in this paragraph will be presented in detail in the incoming stages.

2.1.1 . SICK LIDAR LMS – 500

2.1.1.1 . What is LIDAR?

Lidar stands for Light Detection and Ranging and is very similar to the better known Radar. LIDAR is a better choice than radar, because it has a greater ability to reflect images, making more objects visible. LIDAR uses waves ten to one hundred thousand times shorter than radar waves, which means that it is able to collect much more data. LIDAR is an optical remote sensing technology that can measure the distance to, or other properties of a target by illuminating the target with light, often using pulses from a laser. LIDAR technology has a wide range of applications. The short form LADAR (Laser Detection and Ranging) is often used in military contexts. The term "laser radar" is sometimes used even though LIDAR does not employ microwaves or radio waves and is not therefore in reality related to radar. LIDAR uses ultraviolet, visible, or near infrared light to image objects and can be used with a wide range of targets, including non-metallic objects, rocks, rain, chemical compounds, aerosols, clouds and even single molecules. A narrow laser beam can be used to map physical features with very high resolution. LIDAR has been used extensively for atmospheric research and meteorology. Downward-looking LIDAR instruments fitted to aircraft and satellites are used for surveying and mapping. A recent example is the NASA Experimental Advanced Research Lidar. Wavelengths in a range from about 10 micrometers to the UV are used to suit the target. Typically light is reflected via backscattering.

2.1.1.2 . General/Technical Description

In the following section, there is some technical information about LIDAR and it's architecture. Fig. 2.1. is a screenshot from front view of LIDAR LMS-500.

Fig. 2.1. LIDAR LMS-500

Table 2.1. LIDAR LMS-500 Technical Details

Features	
Model:	LMS-500
Field of application:	Indoor
Version:	Mid Range
Variant:	PRO
Resolution power:	High Resolution
Light source:	Infrared (905 nm)
Laser class:	1 (IEC 60825-1 (2007-6)), eye-safe
Field of view:	190 °
Scanning frequency:	25 Hz / 35 Hz / 50 Hz / 75 Hz / 100 Hz
Angular resolution:	0.167 ° 0.25 ° 0.333 ° 0.5 ° 0.667 ° 1 °
Operating range:	0 m ... 80 m
Max. range with 10 % reflectivity:	26 m
Spot size:	4.7 mrad
Amount of evaluated echoes:	5

Fog correction:	No
Performance	
Response time:	
Detectable object shape:	
Systematic error 1):	± 25 mm (1 m ... 10 m) ± 35 mm (10 m ... 20 m)
Statistical error 2):	± 7 mm (1 m ... 10 m) ± 9 mm (10 m ... 20 m)
Integrated application:	Field evaluation
Number of field sets:	10 fields
Simultaneous processing cases:	10
Interfaces	
Serial (RS-232, RS-422):	YES
Function (Serial (RS-232, RS-422)):	Host
Data transmission rate (Serial (RS-232, RS-422)):	9.6 kBaud ... 500 kBaud
Ethernet:	YES
Function (Ethernet):	Host
Data transmission rate (Ethernet):	10/100 Mbit
Protocol (Ethernet):	TCP/IP, OPC
CAN bus:	YES
Function (CAN bus):	Outputs extension
PROFIBUS DP:	-
PROFINET:	-
DeviceNet:	-
USB:	YES
Function (USB):	AUX
Data transmission rate (USB):	9.6 kBaud ... 500 kBaud
Remark (USB):	Mini-USB
Switching inputs:	4 (Encoder)
Switching outputs:	6
Optical indicators:	5 LEDs (additional 7-segment display)

Mechanics/electronics	
Electrical connection:	1 system plug with screw terminal block
Operating voltage:	24 V DC
Power consumption:	22 W
Housing color:	Light blue (RAL 5012)
Enclosure rating:	IP 65
Protection class:	III
Weight:	3.7 kg
Dimensions:	160 mm x 155 mm x 185 mm
Ambient data	
Object remission:	2 % ... > 1,000 % (reflectors)
Electromagnetic compatibility (EMC):	EN 61000-6-2:2005 / EN 61000-6-3 (2001-10)
Vibration resistance:	EN 60068-2-6 (1995-04)
Shock resistance:	EN 60068-2-27 (1993-03) / EN 60068-2-29 (1993-04)
Ambient operating temperature:	0 °C ... 50 °C
Storage temperature:	-12 °C ... 50 °C
Ambient light safety:	70,000 lx

Used packet rate for LIDAR is 25 Hz in the application.

2.1.1.3 . LIDAR Operating Principles

The LIDAR is an electro-optical laser measurement system that electro-sensitively scans the perimeter of its surroundings in a plane with the aid of laser beams. The LIDAR measures its surroundings in two-dimensional polar coordinates. If a laser beam is incident on an object, the position is determined in the form of distance and direction.

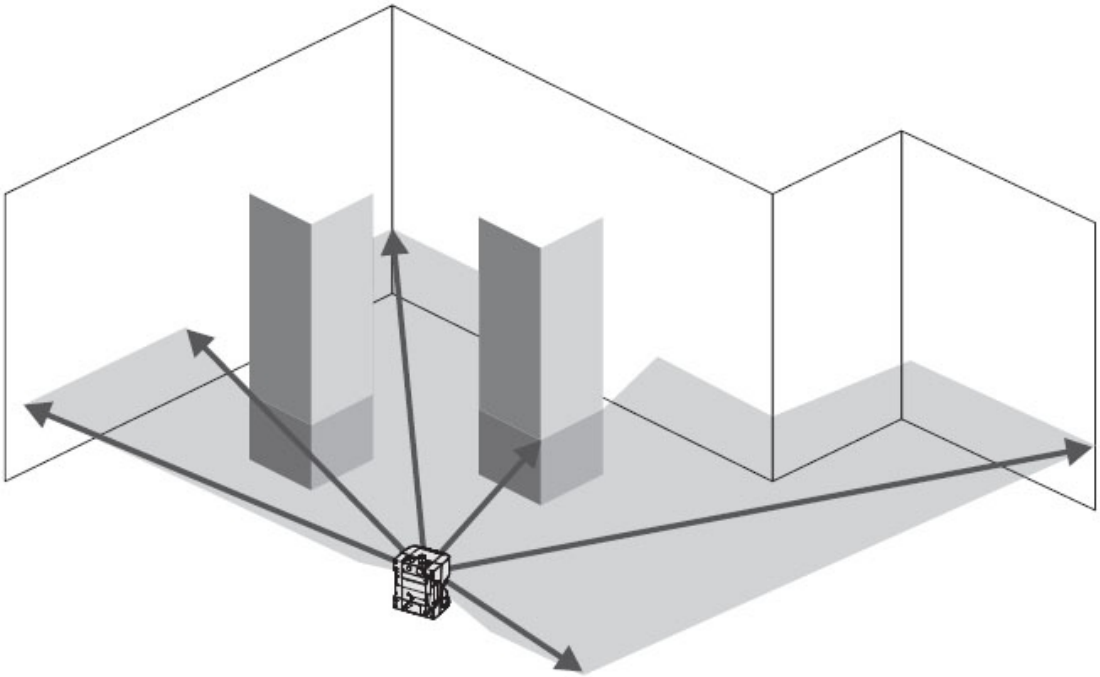

Fig. 2.2. Measuring principle of the LIDAR

Scanning takes place in a sector of 190°. The scanning range of the LIDAR used is maximum 65 m (213.25 ft) on light, natural surfaces with an object remission > 100% (e.g. a white house wall).

Distance measurement

The LIDAR emits pulsed laser beams using a laser diode. If such a laser pulse is incident on an object or a person, it is reflected at its surface. The reflection is detected in the laser
measurement system's receiver using a photodiode.

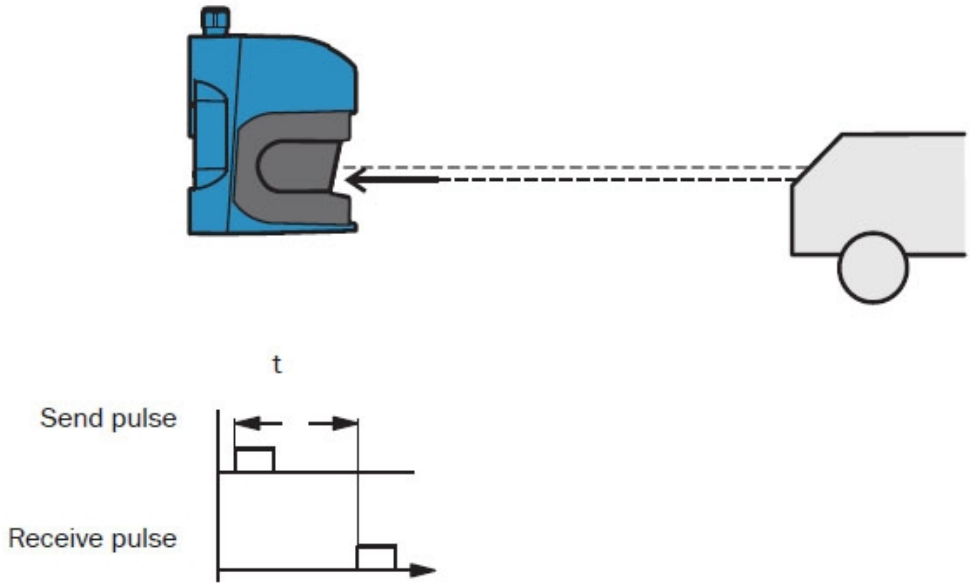

Fig. 2.3. Principle of operation for pulse propagation time measurement

The distance to the object is calculated from the propagation time that the light requires from emission to reception of the reflection at the sensor. This principle of "pulse propagation time measurement" is used by radar systems in a similar manner.

Direction Measurement

The emitted laser beams are deflected using a mirror and scan the surroundings in a circular manner. The measurements are triggered at regular angular steps using an angular encoder.

The LIDAR scans with a scanning frequency of 25, 35, 50, 75 or 100 Hz. During this process, a laser pulse and therefore a measurement is triggered after an angular step of 0.167°, 0.25°, 0.33°, 0.5°, 0.66° or 1°.

Influences of object surfaces on the measurement

The signal received from a perfectly diffuse reflecting white surface corresponds to the definition of a remission of 100%. As a result of this definition, the remissions for surfaces
that reflect the light bundled (mirrored surfaces, reflectors), are more than 100%.

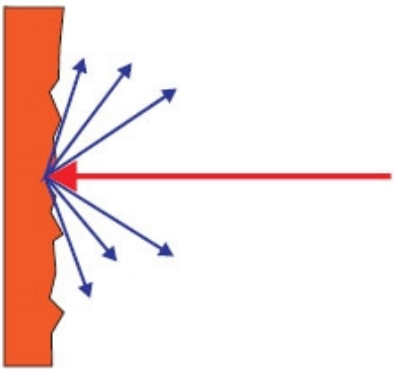

Fig. 2.4. Reflection of the laser beam at the surface of an object

The majority of surfaces reflect the laser beam diffusely in all directions. The reflection of the laser beam will vary as a function of the surface structure and colour. Light surfaces reflect the laser beam better than dark surfaces and can be detected by the LIDAR over larger distances. Brilliant white plaster reflects approx. 100% of the incident light, black foam rubber approx. 2.4%. On very rough surfaces, part of the energy is lost due to shading. The scanning range of the LIDAR will be reduced as a result.

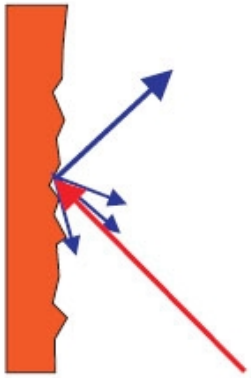

Fig. 2.5. Reflection angle

The reflection angle is the same as the angle of incidence. If the laser beam is incident perpendicularly on a surface, the energy is optimally reflected (Fig. 2.4.). If the beam is incident at an angle, a corresponding energy and scanning range loss is incurred.

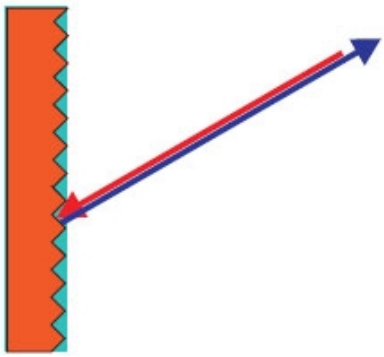

Fig. 2.6. Degree of reflection

If the reflected energy returned is over 100% the incident beam is not reflected diffusely in all directions, but is reflected in a specific direction. As a result a large portion of the energy emitted can be received by the laser distance measurement device. Plastic reflectors (“cats’ eyes”), reflective tape and triple prisms have these properties.

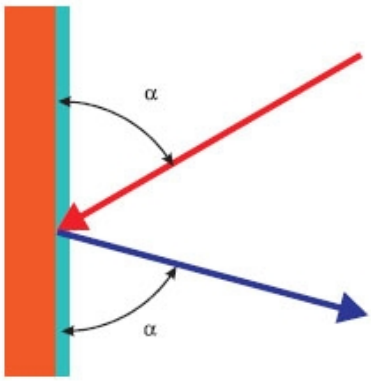

Fig. 2.7. Mirror surfaces

At mirror surfaces the laser beam is almost entirely deflected. Instead of the surface of the mirror, it is possible that the object on which the deflected laser beam is incident may be detected.

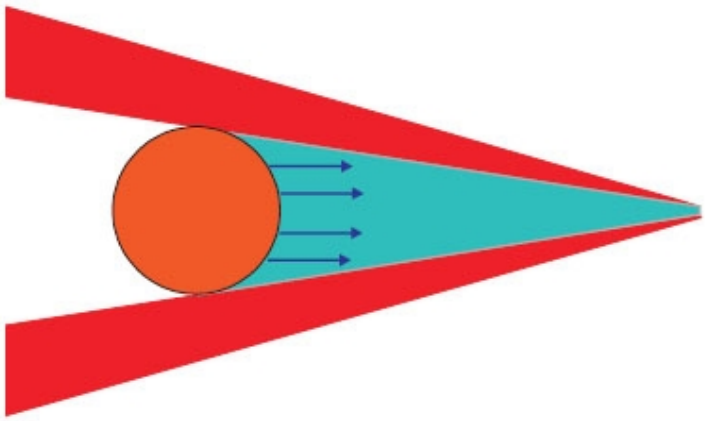

Fig. 2.8. Object smaller than diameter of the laser beam

Objects that are smaller than the diameter of the laser beam cannot reflect all the energy of the laser light. The energy in the portion of the laser light that is not reflected is lost. This means that the scanning range is less than would be possible theoretically based on the surface of the object.

Scanning range of the LIDAR

The scanning range of the LIDAR is dependent on the remission of the objects to be detected. The better a surface reflects the incident radiation, the greater the scanning range of the LIDAR. The diagrams in Fig. 2.9. indicate the relationship between remission and detectability.

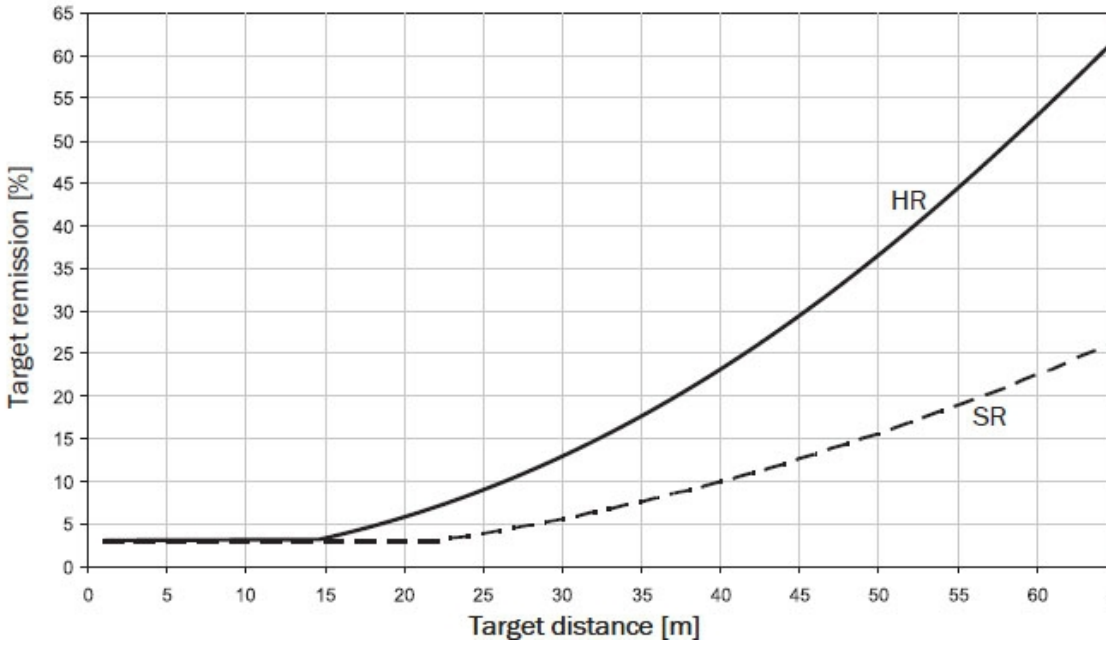

Fig. 2.9. Scanning range of the LIDAR as a function of the target remission

Up to a distance of 10 m (32.81 ft) the LIDAR can detect objects from 3% remission. At a distance of 65 m (213.25 ft) objects are only detected if they have a remission > 100%.

2.1.2 . Crossbow Technology IMU440CA-200

2.1.2.1 . What is IMU (Inertial Measurement Unit) ?

The IMU is a single unit in the electronics module which collects angular velocity and linear acceleration data which is sent to the main processor. The IMU housing actually contains two separate sensors. The first sensor is the accelerometer triad. It generates three analog signals describing the accelerations along each of its axes produced by, and acting on the vehicle. Due to thruster system and physical limitations, the most significant of these sensed accelerations is caused by gravity. The second sensor is the angular rate sensor triad. It also outputs three analog signals. These signals describe the vehicle angular rate about each of the sensor axes. Even though the IMU is not located at the vehicle center of mass, the angular rate measurements are not effected by linear or angular accelerations. The data from these sensors is collected by the IMU 6811 microprocessor through a 12 bit ADC board. The sensor information is then returned to the main processor via a RS422 serial communications interface at a rate of about 200 Hz.

The accelerometer triad, and angular rate sensors within the IMU are mounted such that their sensor coordinate axes are not aligned with those of the vehicle. This is due to the fact that the two sensors in the IMU are mounted in two different orientations in the housing, along with the fact that the axes of the IMU are not aligned with the vehicle axes.

The Figure 2.10. shows two pictures of the IMU, and depicts the direction of the axes of each of the two sensors each with respect to the IMU housing.

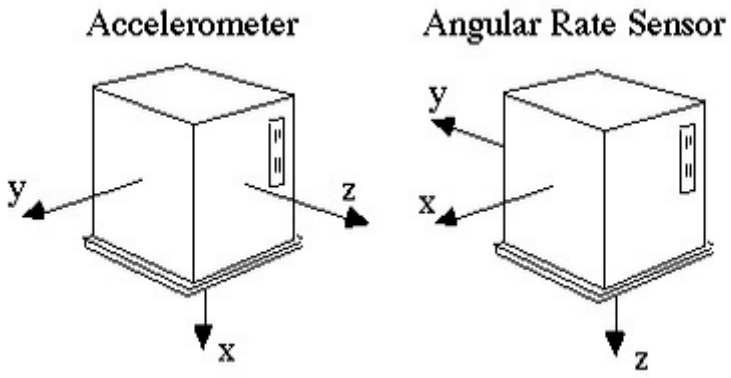

Fig. 2.10. IMU accelerometer and angular axis directions

The IMU440 is offered in standard and high-range sensor configurations and it combines highly-reliable gyros and accelerometers with high-speed DSP electronics to provide a fully calibrated dynamic measurement system in a small and rugged environmentally-sealed enclosure and, with output data also available in delta theta and delta velocity format. It can also provide consistent performance in challenging operating environments and is user-configurable for a wide variety of applications.

2.1.2.2 . General/Technical Specifications

Fig. 2.11. Crossbow IMU440CA-200

Table 2.2. Crossbow IMU440CA-200 Technical Details

General Specifications

Gyro Range	200
In-Run Stability	10
Non-Linearity	1
Bandwidth	25
Accel Range	4
In-Run Stability	1
Non-Linearity	1
Bandwidth	25

Performance

Angulare Rate

Range: Roll, Pitch, Yaw (°/sec)	± 200 (± 400 option available)
Bias Stability In-Run (°/hr)	< 10
Bias Stability Over Temp (°/sec)	< 0.2
Scale Factor Accuracy (%)	< 1
Non-Linearity (% FS)	< 0.5
Resolution (°/sec)	< 0.02
Angle Random Walk (°/√hr)	< 4.5
Bandwidth (Hz)	25

Acceleration

Input Range: X/Y/Z (g)	± 4 (± 10 option available)
Bias Stability In-Run (mg)	< 1
Bias Stability Over Temp (mg)	< 4
Scale Factor Accuracy (%)	< 1
Non-Linearity (% FS)	< 1
Resolution (mg)	< 0.5
Velocity Random Walk (m/s/√hr)	<1.0
Bandwith (Hz)	25

Specifications

Environment

Operating Temperature (°C)	-40 to +71
Non-Operating Temperature (°C)	-55 to +85
Enclosure	IP66 Compliant

Electrical

Input Voltage (VDC)	9 to 42
Power Consumption (W)	< 3
Digital Interface	RS-232

Physical

Size	(in)	3 x 3.75 x 2.50
	(cm)	7.62 x 9.53 x 6.43
Weight	(lbs)	< 1.2
	(kg)	< 0.55
Connector		DB15, D-sub 15-pin Male

Used packet rate for LIDAR is 25 Hz in the application.

2.1.3 . Roboteq AX 1500 Motor Controller & Odometer

Fig. 2.12. Roboteq AX1500 mainboard

Roboteq's AX1500 controller is designed to convert commands received from a R/C radio, Analog Joystick, wireless modem, or microcomputer into high voltage and high current output for driving one or two DC motors.

The controller's two channels can either be operated independently or mixed to set the direction and rotation of a vehicle by coordinating the motion on each side of the vehicle. The motors may be operated in open or closed loop speed mode. Using low-cost position sensors, they may also be set to operate as heavy-duty position servos.

Table 2.3. Roboteq AX1500 Technical Details

Technical Features

Power Stage

Operating Voltage	12V to 40V DC	
Number of Channels	2	
Max Current	30A	40A
30s	25A	35A
1min	20A	30A
3min	20A	30A
1h		
Surge Current	>150A	
MOSFETs per Channel	4	
ON Resistance	10 mOhm	
Current Limiting	By automatic power output reduction according to user preset limit and temperature	
Synchronous Rectification	Yes - Allows regenerative braking	
Temperature protection	Automated current limit reduction starting at 80o C (175o F) heat sink temperature	
Voltage protection	Output shut off below 12V and above 43V	
Power Wiring	Terminal strip. AWG 12 max cable	

Command

R/C Inputs	2 + 1 accessories (1.0ms - 1.5ms center - 2ms, Adjustable)
Serial Interface	RS232. 9600 bauds
Analog Interface	2 inputs (0V - 2.5V center - 5V)

Input Corrections	Ch1 & Ch2 mixing for tank steering. Programmable deadband. 4 Exponent & Logarithmic command curves.

Input/Outputs

Optical Encoder Interface	Yes
Analog Inputs	4 inputs, 8-bit resolution
Digital Outputs	1 output, 24V 1A max
Digital Inputs	up to 3 general purpose inputs
5V Supply Output	100mA max for Radio or other devices

Operating Modes	
Open Loop Speed	Forward & Reverse Speed Control. Separate or Mixed
Closed Loop Speed	Use Tachometer on analog inputs & PID
Position Mode	Use Potentiometer on analog inputs & PID
Controller Configuration	Jumper-less using PC utility

Phsyical

Operating Temperature	-40 to +85oC heat sink temperature
Enclosure	Unenclosed, board level
Controller size	4.2" (106mm) wide x 4.2" (106mm) long x 1.5" tall (38mm) tall including heat sink
Cables	10" (25cm) RC cable to Radio. 4' (1m) RS232 cable for PC connection
Weight	3 oz (85g)

2.1.4 . SiRFstarIII GPS Receiver

2.1.4.1 . What is GPS?

The Global Positioning System(GPS) is one of the key components of the project. GPS is used to find absolute position and velocity. It is a satellite-based navigation system made up of a network of 24 satellites placed into orbit by the U.S. Department of Defense. GPS was originally intended for military applications, but in the 1980s, the government made the system available for civilian use. GPS works in any weather conditions, anywhere in the world, 24 hours a day. There are no subscription fees or setup charges to use GPS.

GPS satellites circle the earth twice a day in a very precise orbit and transmit signal information to earth. GPS receivers take this information and use triangulation to calculate the user's exact location. Essentially, the GPS receiver compares the time a signal was transmitted by a satellite with the time it was received. The time difference tells the GPS receiver how far away the satellite is. Now, with distance measurements from a few more satellites, the receiver can determine the user's position and display it on the unit's electronic map. Today's GPS receivers are extremely accurate, thanks to their parallel multi-channel design.

GPS satellites transmit two low power radio signals, designated L1 and L2. Civilian GPS uses the L1 frequency of 1575.42 MHz in the UHF band. The signals travel by line of sight, meaning they will pass through clouds, glass and plastic but will not go through most solid objects such as buildings and mountains.

A GPS signal contains three different bits of information - a pseudorandom code, ephemeris data and almanac data. The pseudorandom code is simply an I.D. code that identifies which satellite is transmitting information. Ephemeris data, which is constantly transmitted by each satellite, contains important information about the status of the satellite (healthy or unhealthy), current date and time. This part of the signal is essential for determining a position. The almanac data tells the GPS receiver where each GPS satellite should be at any time throughout the day. Each satellite transmits

almanac data showing the orbital information for that satellite and for every other satellite in the system.

There are some factors that can degrade the GPS signal and thus affect accuracy include the following:

1) Ionosphere and troposphere delays:
The satellite signal slows as it passes through the atmosphere. The GPS system uses a built-in model that calculates an average amount of delay to partially correct for this type of error.

2) Signal multipath:
This occurs when the GPS signal is reflected off objects such as tall buildings or large rock surfaces before it reaches the receiver. This increases the travel time of the signal, thereby causing errors.

3) Receiver clock errors:
A receiver's built-in clock is not as accurate as the atomic clocks onboard the GPS satellites. Therefore, it may have very slight timing errors.

4) Orbital errors:
Also known as ephemeris errors, these are inaccuracies of the satellite's reported location.

5) Number of satellites visible:
The more satellites a GPS receiver can "see," the better the accuracy. Buildings, terrain, electronic interference, or sometimes even dense foliage can block signal reception, causing position errors or possibly no position reading at all. GPS units typically will not work indoors, underwater or underground.

6) Satellite geometry/shading:
This refers to the relative position of the satellites at any given time. Ideal satellite

geometry exists when the satellites are located at wide angles relative to each other. Poor geometry results when the satellites are located in a line or in a tight grouping.

7) Intentional degradation of the satellite signal:
Selective Availability (SA) is an intentional degradation of the signal once imposed by the U.S. Department of Defense. SA was intended to prevent military adversaries from using the highly accurate GPS signals. The government turned off SA in May 2000, which significantly improved the accuracy of civilian GPS receivers.

2.1.4.2 . General/Technical Details

Fig. 2.13. SiRFstarIII GPS Receiver

Table 2.4. SiRFstarIII GPS Receiver Technical Details

Technical Details

Horizontal Position Accuracy	• Autonomous <2.5 m • SBAS <2.0m
Velocity Accuracy	• Speed <0.01 m/s • Heading <0.01
Time To First Fix	• Hot start - Autonomous <1 s • Warm start - Autonomous <35 s • Cold start - Autonomous <35 s • With SiRInstantFix - as low as 5 s
Sensitivity	• Autonomous acquisition -142 dBm • Tracking -159 dBm
Receiver	• Tracking - L1, CA Code • Channels - up to 20 • Max update rate - 1 Hz • Max altitude/velocity - <60,000 ft/<1,000 knots • Protocol support - SiRF Binary, NMEA
System Integration	• I/O Interface - UART • External reference clock - 16.369 and 26 MHz • RTC input - 32.768 kHz
Power	• Continuous Autonomous operation - 62 mW • TricklePower - 40 mW
Size	10 x 10 x 1.4 mm

2.1.5 . Entire System Overview

When the system is observed as a whole, one can find a metal platform, located at the base of the system and attached to the wheels which allow movements, that carries all the components involved. The IMU and the Odometer are placed on the interior part, that is not visible in the photographs, of the metal platform; and just above this division there exists 3 power supplies which are connected to all of the components in the system. LIDAR is located at the top of the system and the GPS is placed in the open area at the front of the system in order for it to establish connections with the satellites in the most efficient and effective way. As the system does not incorporate any embedded software, enough space is reserved at the front section of the system for the computer from which the software will be run. The design of the system are shown in the pictures below.

Fig. 2.14. A screen-shot of system design from front view

Fig. 2.15. A screen-shot of system design from back view

Fig. 2.16. A screen-shot of IMU

2.2 . Main Steps

Out of numerous different options, in our approach, the main component used is LIDAR because of the initial need to detect the union of points reflected from objects which are then grouped together and named as a “cluster”, after applying different algorithms. After receiving the reflected points which are received from LIDAR as a pair of degrees (between -5 / 185) and distance, 3 main steps are performed in order to achieve a list of most valuable clusters, assuming that the environment is urban, road-network based or with many pedestrians:

1- Place each point in clusters to form a group of returns that are within a specified distance from each other. If a point is not within the specified distance from a point in the currently calculated cluster, then open a new cluster and place the point into the newly opened cluster. Each cluster is assigned a unique ID.

2- Find a centroid for each cluster; calculate the distance between that centroid and each cluster centroid. If the calculated distance is less than the threshold, get the combination of these two clusters as they will form a single cluster.

3- Find the points which are most distant to each other for each cluster; calculate this distance as an Euclidian distance; and eliminate the clusters which are less than the threshold. Also, eliminating clusters which have less than a threshold number of points can sometimes give more accurate results.

On the other hand, getting data from LIDAR and clustering the points are not sufficient in terms of detecting dynamic objects when LIDAR's position is fixed. The movement of each cluster must also be estimated considering the environment and nature of the objects to be tracked. It is important to notice that we know nothing but the objects returned by LIDAR. Thus, the result is not 100% accurate but the main goal is achieving the maximum accuracy. With these considerations, a minimum-maximum movement of a cluster's centroid in a particular time interval is calculated, giving us an idea about the possible trajectory of an object. Afterwards, some time-related data

about different clusters are obtained.

After the stage of object detection, object tracking must be performed in order to match clusters in a time with the clusters in the previous time. To do that, each cluster found in time t is compared with the clusters found in time t+1 by analyzing the centroid displacement, the possible path of the object and the cluster's start-end angles. All of these calculations are valid where the position of the LIDAR is fixed. Where LIDAR's position is not fixed, the use of the IMU, Odometer and Particle Filter Algorithm is obligatory. Use of GPS can be optional but it is also added to the system in order to increase accuracy.

As stated above, the data received from LIDAR is insufficient without finding the relative position changes when comparing/rotating already found clusters before comparing with the ones in the previous time. Therefore, the IMU (Inertial Measurement Unit) is used to calculate the acceleration and rotation of the observer. Especially when counting the cumulative error tolerance of the acceleration, Odometer data is needed which is used in particular with the IMU data in the Particle Filter Algorithm. Knowing that the Odometer data gives the rotation changes in a given time interval, a more accurate result is obtained when it is in use with the IMU data. GPS data is also added to the Particle Filter Algorithm to help reducing the cumulative error tolerance of the IMU.

The figure 2.17. represents the entire data flow of the system:

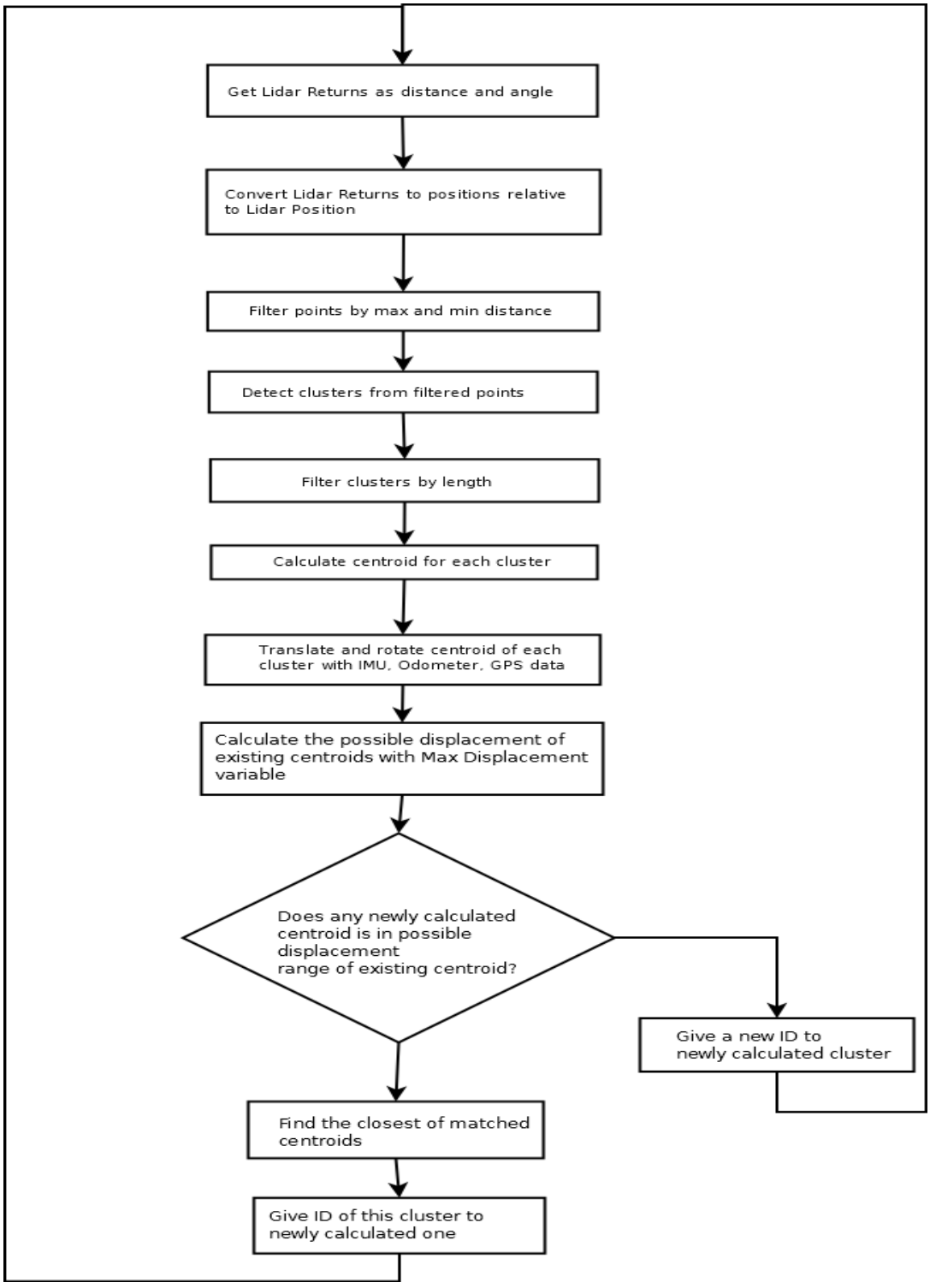

Fig. 2.17. Entire system data flow

3 . Fusion and Detection Methodology

3.1 . General Concept

The whole object detection and tracking implementation is done by 6 general steps as follows:

1- Getting LIDAR returns,
2- Detecting clusters from LIDAR returns,
3- Normalizing existing clusters with IMU, Odometer and GPS data,
4- Implementation of the Particle Filter Algorithm,
5- Comparison and identification of clusters
6- Repeat steps 1-5 until the end of program execution

Before the clustering period, some environment variables must be sent, according to the type of object, to track distance, possible movement and object structure. The main environment variables are minimum range, maximum range, minimum length of the cluster, maximum length between two points in a cluster and maximum possible displacement of the cluster's centroid.

Each step of the algorithm is explained in the following section of the thesis.

3.2 . Getting LIDAR Returns

The whole operation starts with receiving the data from LIDAR. The data is a subset of the distance and the angle, which gives us the relative position of the points from LIDAR's position. Having received the data set for all points, we first filter the points by considering the minimum and maximum distances that we want to examine. Although the LMS-500 LIDAR generates data between -5º and 185º, we only use the

range between 0º and 180º, in order to not implicate the output of the application. LMS 500's resolution of the angular step width is a minimum of 0.168o; hence we will be in contact with 1,137 points in each LIDAR return, with a scanning frequency of 25 Hz. To decrease the CPU load, we will implement the clustering at 10 Hz.

3.3 . Detecting Clusters from LIDAR Returns

After receiving LIDAR returns, before doing the clustering, LIDAR data subsets must be converted to a global coordinate system with the following equations where (0,0) is the position of LIDAR :

$$\text{LIDAR data subset} = (d_i, \alpha_i) \tag{3.1}$$

$$x_i = d_i \cos(\alpha_i) \tag{3.2}$$

$$y_i = d_i \sin(\alpha_i) \qquad for\ i=1...N \tag{3.3}$$

The clustering process starts with the decision of whether a point is in a point union (cluster) or not. In doing so, the minimum cluster length, and the maximum length between two points in a cluster must be taken into account. In addition to these considerations, there must also be a minimum tolerance $Min(d_1)$ for the distance between two points in a cluster due to the fact that the distance and the angle resolution of LIDAR affect the possible minimum distance for the consecutive points.

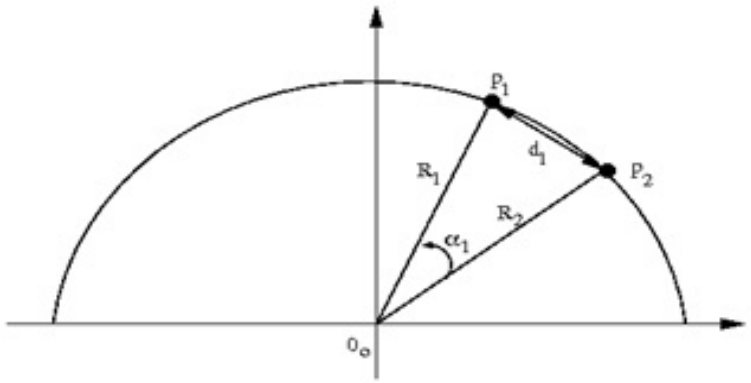

Fig. 3.1. Shows how the calculation is done for minimum possible distance between two consecutive points of LIDAR data

R_1, R_2 = Distance from LIDAR to P1 and P2 respectively

d_1 = Distance between P1 and P2

The minimum distance d_1 which is named as tolerance is calculated as follows:

LIDAR minimum angular resolution is:

$$Min(\alpha)=0.168 \tag{3.4}$$

$$Min(d_1)=2\sin(0.084) \tag{3.5}$$

If the distance from a point to any point in a cluster is less than or equal to the default tolerance plus the maximum possible length between two points in a cluster, the distance is calculated as d_1 in Fig. 3.1. This intermediate point is added into the candidate cluster and this step is repeated until every point in a filtered LIDAR measurement is processed.

After grouping points together as clusters, we calculate a centroid for all of the clusters by:

$$C_x = \frac{\sum x_i}{N} \quad C_y = \frac{\sum y_i}{N} \text{ for } i=1....N \tag{3.6}$$

In the third stage, each cluster is filtered by their length, which is found by calculating two points that are furthest away from each other in a cluster. If the calculated distance is less than the threshold, the cluster is eliminated.

Last stage of the clustering is the joining of the clusters, which have centroids that are closer to each other than the threshold. If the distance from the centroid of a cluster to another is less than the half of the maximum length between two points in a cluster, two clusters are combined. The purpose of this stage is to eliminate the clusters which are very close to each other in order to not to adversely affect the cluster comparison and identification stage.

3.4 . Normalizing Existing Clusters with IMU and Odometer Data

In order to calculate relative movements of each cluster from time t to t+1, LIDAR's relative position must be known or predicted. The relative position of LIDAR can be calculated by the use of the IMU (Inertial Measurement Unit), which gives the x,y,z accelerations and x,y,z yaw rates. In our situation, we assume that there is no change in x and y yaw rates and we only take z yaw rate into account. Due to the noticeable noise especially in the acceleration data, the Particle Filter Algorithm is integrated into the project, which requires also another real time movement data in our system, the Odometer and the GPS. The calculations are done in the Implementation of Particle Filter.

3.5 . Implementation of Particle Filter

Use of the Particle Filter Algorithm allows us to estimate recursively the displacement of the LIDAR’s position. The main goal is to track a moving vehicle whose motion can be described as a dynamic system. State and measurement model of the vehicle can be described as follows:

$$x_t = A\,x_{t-1} + B\,u_{t-1} + w_{t-1} \tag{3.7}$$

$$z_t = h(x_{t-1}) + e_{t-1} \tag{3.8}$$

where $x_t, u_{t-1}, w_{t-1}, z_t, e_{t-1}$ denotes the state vector, measured inputs, faults, measurement vector and measurement error respectively. Tracking state is represented as vector $x = [C_x, C_y, \alpha_i]$ where the three arguments present the centroid's coordinates, orientation and it's range, respectively. The vectors u is the wheel odomoter and z represents the measurements of LIDAR and IMU sensors.

- Setting up the general system equation

$$x_{t+1} = Ax_t + Bu_t + w_t \tag{3.9}$$

where A and B are identity matrix and w_t is the noise.

$$x_{t+1} = x_t + u_t + w_t \tag{3.10}$$

where u_t is the value calculated from wheel odometer as follows:

$$u_t = \int_{t-1}^{t} RPM_t * R_{tire} * \pi \tag{3.11}$$

where RPM_t is rounds per minute at time t and R_{tire} is the diameter of vehicle tire.

- Initialization of the particles x_i for *i=1....N* where *N=1000*

- Translation of all particles by using IMU z yaw rate and x,y acceleration (z acceleration is taken into account while calculating x,y acceleration where the environment is not flat)

$$\Delta x_{dep} = \iint_{t}^{t+1} a \tag{3.12}$$

where a is the acceleration data coming from IMU.

$$z_t^i = x_{t-1}^i + \Delta x_{dep} \text{ for } i=1....N \tag{3.13}$$

- A weight is calculated for each of the particles by the use of normal distribution as follows:

$$w_t^i = w_{t-1}^i \, p\langle z(t) | x(t)_i \rangle \text{ for } i=1....N \tag{3.14}$$

- Weight normalization and position estimation is given by:

$$X_{est}(t) = \sum w_t^i x_t^i \tag{3.15}$$

$$Y_{est}(t) = \sum w_t^i y_t^i \text{ for } i=1....N \tag{3.16}$$

- An effectiveness value is calculated in every step to sustain the calculation accuracy:

$$N_{eff} = \frac{1}{\sum (w_t^i)^2} \text{ for } i=1.....N \tag{3.17}$$

If N_{eff} < 0.25 then particles which have small weights are replaced with the bigger weights.

Every LIDAR return will be translated with the estimation obtained from the Particle Filter and rotated with the value received from the IMU, which has a great accuracy in comparison with the IMU acceleration data.

3.6 . Comparison and Identification of Clusters

In order to match the cluster with the existing ones and decide which of them are the same as the previous ones, following actions need to be taken. Firstly, a cluster array for a specified time interval is required. This will help us to match a cluster which has just disappeared and then quickly reappeared. In case there is not any cluster array, the matching becomes more and more inaccurate, keeping in mind the noise factor and the possibility of two different objects falling at the same LIDAR angle for a short period of time. Besides, the distance between every cluster centroid and the ones in the history are checked by considering the maximum possible displacement in a second and the elapsed time between the formation of the compared clusters. If it is found that the distance from the centroid of a new cluster to another cluster, taken from cluster array, is less than the maximum possible displacement, the corresponding cluster is added to a subset of clusters. We repeat the same operation for every cluster in the cluster array and we take the one which minimizes the distance in this subset. Once a cluster is matched with another, all the clusters with the same IDs as the matched one are eliminated from the cluster array, in order to prevent matching the same object with the same ID to more than one object. If no cluster is found for matching, a newly generated incremental unique ID is given to the cluster.

4 . Experimental Results

4.1 . Test Conditions, Metrics and Summary

We conducted several experiments to evaluate our implementation in different environments. SICK LMS-500 LIDAR is used for the tests. The general test purpose is to check the accuracy of cluster detection in different environments. Cluster centroids are the key factors of the tracking and need to be focused on in order to obtain better results.(Fig. 4.1.)

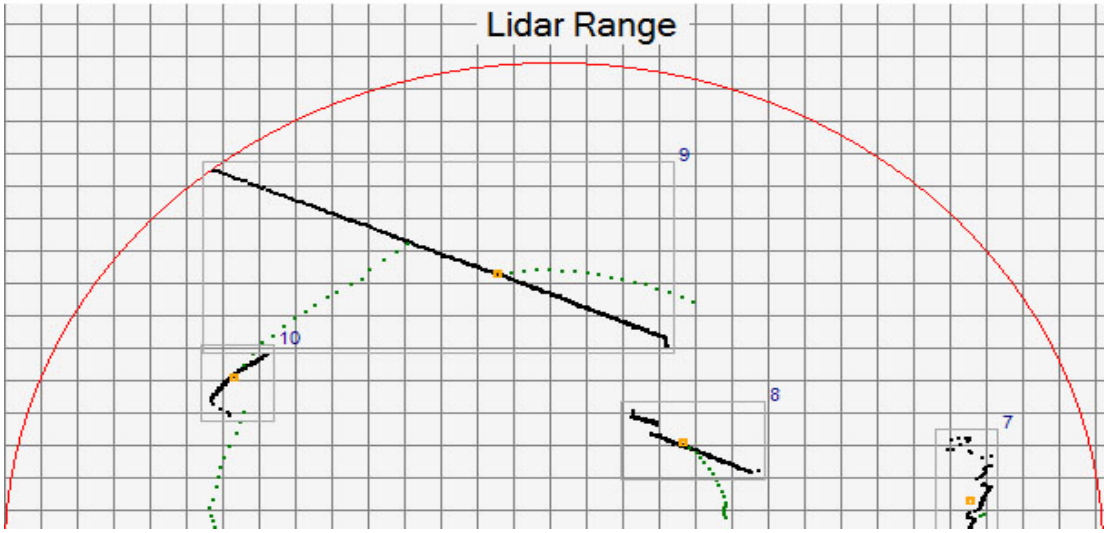

Fig. 4.1. Shows a screen-shot from centroid changes of 4 different fixed objects after 70 degree clockwise rotation of the Lidar. IMU yaw rate data is used to rotate the existing cluster centroids.

The tests show in summary that when LIDAR is mobile the tracking accuracy is decreased significantly. The tracking accuracy is also decreased with increasing max range. In tight areas, excessive number of clusters affects the tracking as seen when comparing the Table 4.1 and Table 4.3. It can be also detected from Table 4.3 and Table 4.4. that detection accuracy is decreased while measurement time interval is increased

because of possible collisions of detected objects and possible cumulative error and noise. The best results are obtained with fixed LIDAR position, minimum range and minimum measurement time interval.

Table 4.1. Detection accuracy in summary for fixed positions of clusters and LIDAR scanner in 15 seconds interval.

LIDAR and Clusters have fixed positions (Time Interval 15 seconds)				
	Clusters			
Max Range(mm)	Observed	Detected	Successfully tracked	Accuracy
1500	1	1	1	100,00%
5000	1	1	1	100,00%
10000	3	3	3	100,00%
30000	3	3	3	100,00%

Table 4.2. Detection accuracy in summary for fixed positions of clusters and LIDAR scanner in 30 seconds interval.

LIDAR and Clusters have fixed positions (Time Interval 30 seconds)				
	Clusters			
Max Range(mm)	Observed	Detected	Successfully tracked	Accuracy
1500	1	1	1	100,00%
5000	1	1	1	100,00%
10000	3	3	3	100,00%
30000	3	3	2	66,60%

Table 4.3. Detection accuracy in summary for fixed LIDAR and dynamic Clusters in 15 seconds interval.

LIDAR has fixed position, Clusters are mobile (Time Interval 15 seconds)				
	Clusters			
Max Range(mm)	Observed	Detected	Successfully tracked	Accuracy
1500	4	4	3	75,00%
10000	4	4	3	75,00%

30000	6	5	2	33,33%

Table 4.4. Detection accuracy in summary for fixed LIDAR and dynamic Clusters in 30 seconds interval.

LIDAR has fixed position, Clusters are mobile (Time Interval 30 seconds)				
	Clusters			
Max Range(mm)	Observed	Detected	Successfully tracked	Accuracy
1500	4	4	3	75,00%
10000	4	4	2	50,00%
30000	6	5	1	16,66%

Table 4.5. Detection accuracy in summary for mobile LIDAR and dynamic Clusters in 15 seconds interval.

LIDAR and Clusters are mobile *(Time Interval 15 seconds)*				
	Clusters			
Max Range(mm)	Observed	Detected	Successfully tracked	Accuracy
1500	3	3	2	66,66%
10000	6	5	3	50,00%
30000	6	5	1	16,66%

Table 4.6. Detection accuracy in summary for mobile LIDAR and dynamic Clusters in 30 seconds interval.

LIDAR and Clusters are mobile *(Time Interval 30 seconds)*				
	Clusters			
Max Range(mm)	Observed	Detected	Successfully tracked	Accuracy
1500	3	3	2	66,66%
10000	6	5	2	33,33%
30000	6	5	0	0,00%

To better evaluate test results, the performance of detection and tracking is calculated by

the following metrics: Detection Rate, Tracking Rate, Range Detection Rate, Range Tracking Rate, Time Detection Rate, Time Tracking Rate .

DR (Detection Rate) is the rate of detected objects per total objects and is calculated as follows:

$$DR = \frac{detected\ objects}{total\ objects} \tag{4.1}$$

TR (Tracking Rate) is the rate of successfully tracked objects per detected objects and is calculated as follows:

$$TR = \frac{successfully\ tracked}{detected\ objects} \tag{4.2}$$

RDR (Range Detection Rate) is the rate of undetected of objects per mm and is calculated is as follows:

$$RDR = \frac{1 - DR}{range(mm)} \tag{4.3}$$

RTR (Range Tracking Rate) is the rate of untracked objects per mm and is calculated as follows:

$$RTR = \frac{1 - TR}{range(mm)} \tag{4.4}$$

TDR (Time Detection Rate) is the rate of undetected objects per second and is calculated as follows:

$$TDR = \frac{1 - DR}{elapsed\ time(sec)} \tag{4.5}$$

TTR (Time Tracking Rate) is the rate of untracked objects per sec and is calculated as follows:

$$TTR = \frac{1 - TR}{elapsed\ time(sec)} \tag{4.6}$$

4.2 . Tests

4.2.1 . Range Test

Range tests are applied for 30 seconds time interval with dynamic clusters and fixed LIDAR.

Table 4.7. Range Test

	Rates			
Max Range(mm)	DR	TR	RDR	RTR
1500	1	1	0	0
10000	1	0.88	0	0.000012
20000	0.9	0.72	0.000005	0.000014

In range tests, increasing RDR shows that rate of failed detections increases with increasing range which means that range increases possible detection error rate. Increasing RTR shows that rate of failed trackings increases with time which means that with increasing range, tracking error rate becomes higher.

4.2.2 . Time Test

Time tests are applied in 10000 mm range with dynamic clusters and fixed LIDAR.

Table 4.8. Time Test

Time(sec)	Rates			
	DR	TR	RDR	RTR
30	1	0.88	0	0.0040
60	0.97	0.83	0.00005	0.0028
120	0.90	0.82	0.00083	0.0015

In time tests, increasing TDR shows that rate of failed detections increases with increasing time which means that possible detection error rate is increasing with time. Decreasing TTR shows that rate of failed trackings decreases with time which means that possible tracking error rate becomes smaller.

4.2.3 . Test with Fixed LIDAR Position + Fixed Clusters

Two tests are done with fixed LIDAR and cluster position from different views and with different environment variables.

Environment variables used for the application are:

Min range: 22mm,
Max range: 12000mm,
Minimum cluster length: 800mm,
Max length between 2 points in a cluster: 90mm,
Max displacement: 50mm/sec

Fig. 4.2. A camera view of a dynamic environment which will be analyzed with LIDAR and clustered

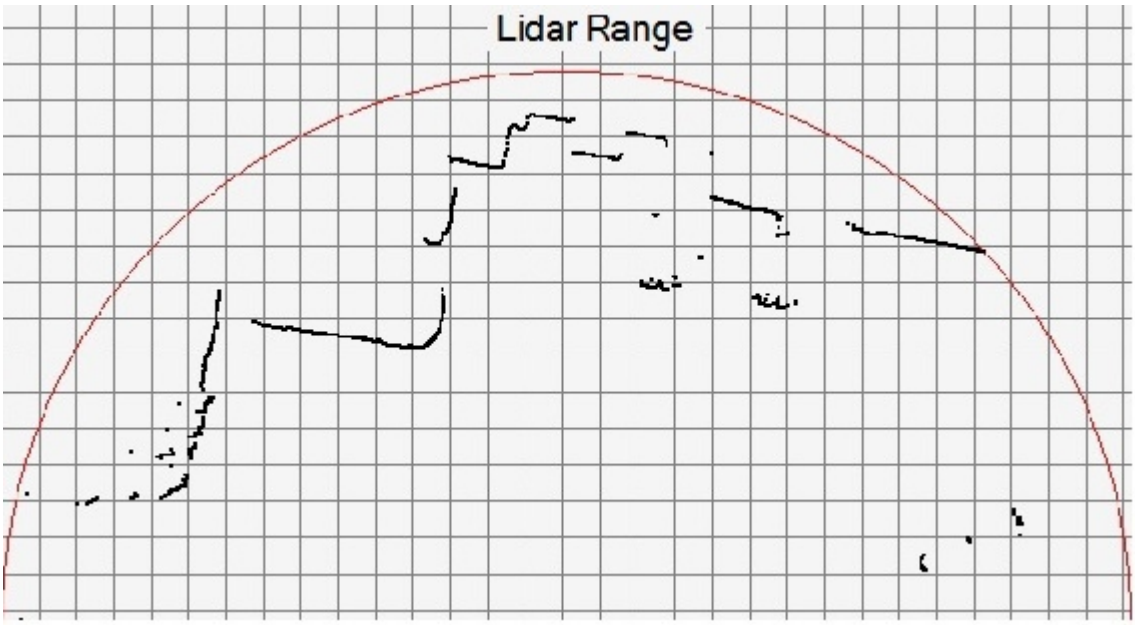

Fig. 4.3. Non-clustered view of Figure 4.2.

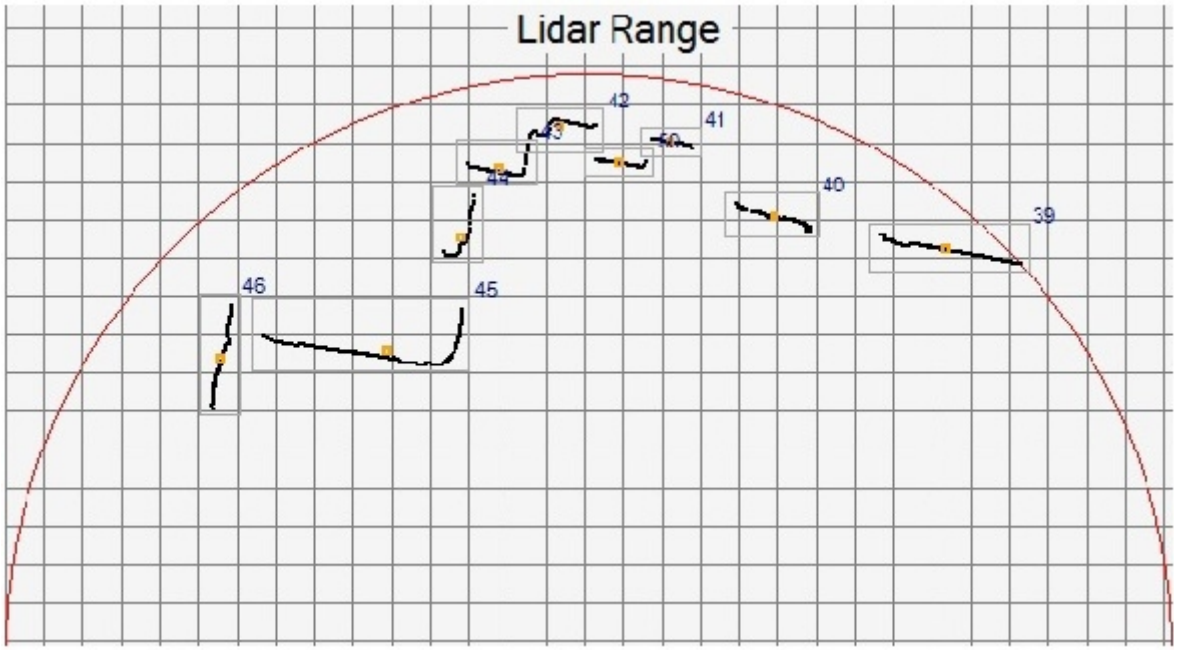

Fig. 4.4. Clustered view of Figure 4.2.

For the second test in the same environment after changing LIDAR position, here are the environment variables used:

Min range: 22mm,
Max range: 12000mm,
Minimum cluster length: 1500mm,
Max length between 2 points in a cluster: 150mm,
Max displacement: 50mm/sec

Fig. 4.5. A camera view of the second test from a dynamic environment

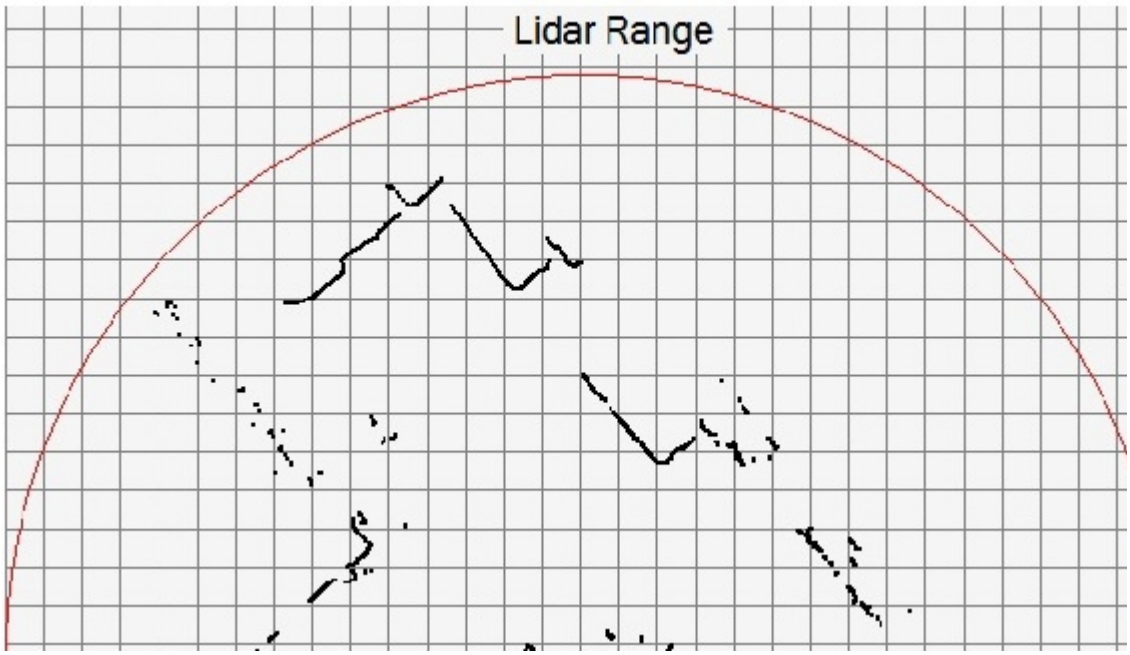

Fig. 4.6. Non-clustered view of Figure 4.5.

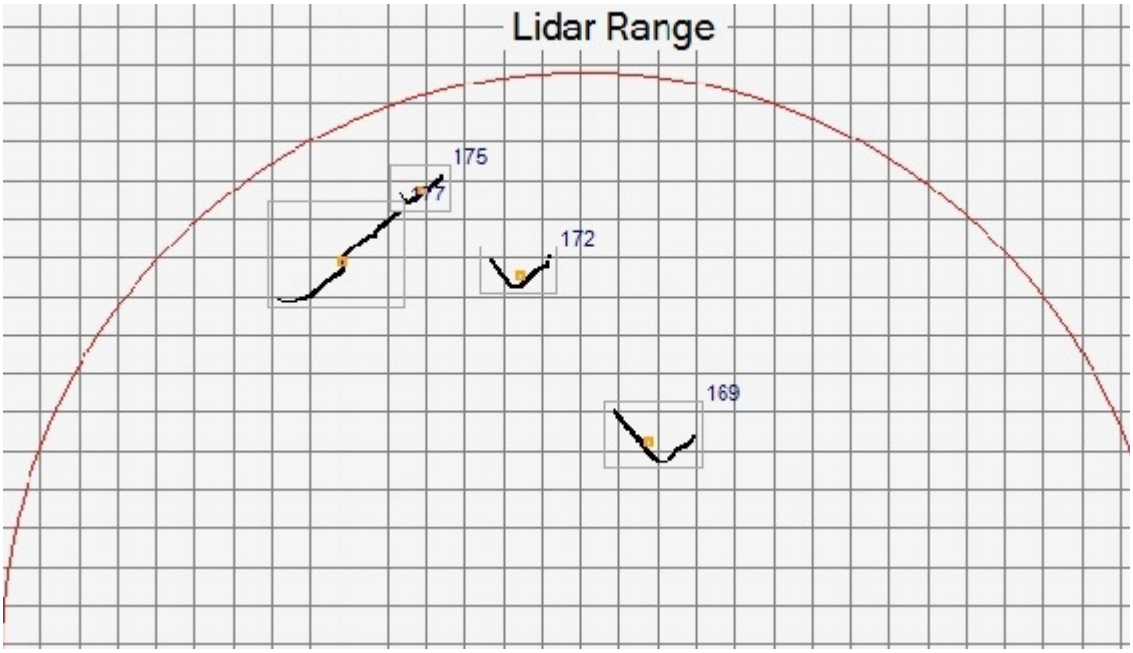

Fig. 4.7. Clustered view of Figure 4.5.

Minimum cluster length for clustering LIDAR returns for a car must adopt the car dimensions while it is different when detecting walking humans. In the case where we only want to filter car-shaped objects, minimum cluster length must be larger than 1500mm taking into account the two most distant points in a car cluster. Second test is done for detecting and tracking multiple walking humans.

4.2.4 . Test with Fixed LIDAR Position + Dynamic Clusters (Human)

Two tests are done with fixed LIDAR and moving humans. First test is with a single walking human while the second is with two walking humans. The main difference to detect a walking human from detecting a car is that while walking, length between legs are increased and this cause the need of increasing maximum length between 2 points in a cluster in comparison with car clustering. Also, as the tracked objects don't have fixed positions, their maximum possible displacement must be as higher as 1500mm/sec. These tests are done especially with a maximum range of 1500mm to check the detection and tracking accuracy in tight areas.

The environment variables for this test are:

Min range: 22mm,

Max range: 1500mm,

Minimum cluster length: 300mm,

Max length between 2 points in a cluster: 150mm,

Max displacement: 1500mm/sec

The following Fig. 4.8 and Fig. 4.9 are the outputs from an environment with single walking human:

Fig. 4.8. Clustered view of a moving human in a dynamic environment

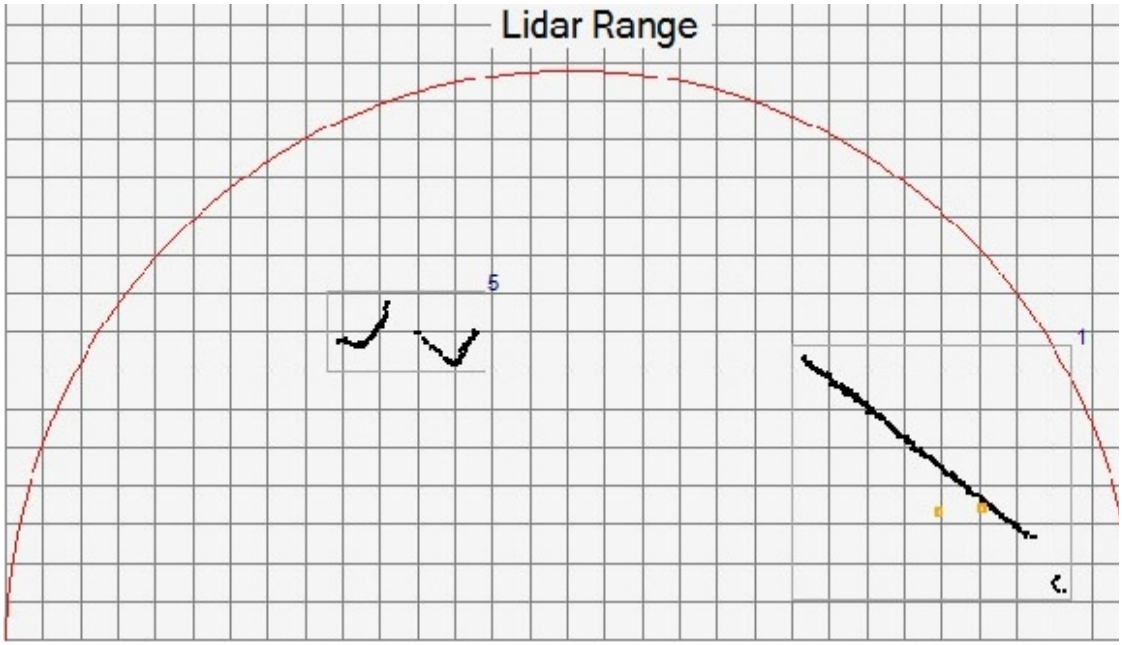

Fig. 4.9. Clustered view of a moving human after 2 seconds from screenshot time of Figure 4.8.

The second test is done with two humans. Test is kept going for 30 seconds while two humans continue walking without colliding and every one in his area. The humans are successfully clustered. If the moving clusters fall into the 150 mm range from each other's centroid and keep their position fixed for more than a second, they are classified as one cluster and their ID is changed. The results are shown in Fig. 4.10. and Fig. 4.11.

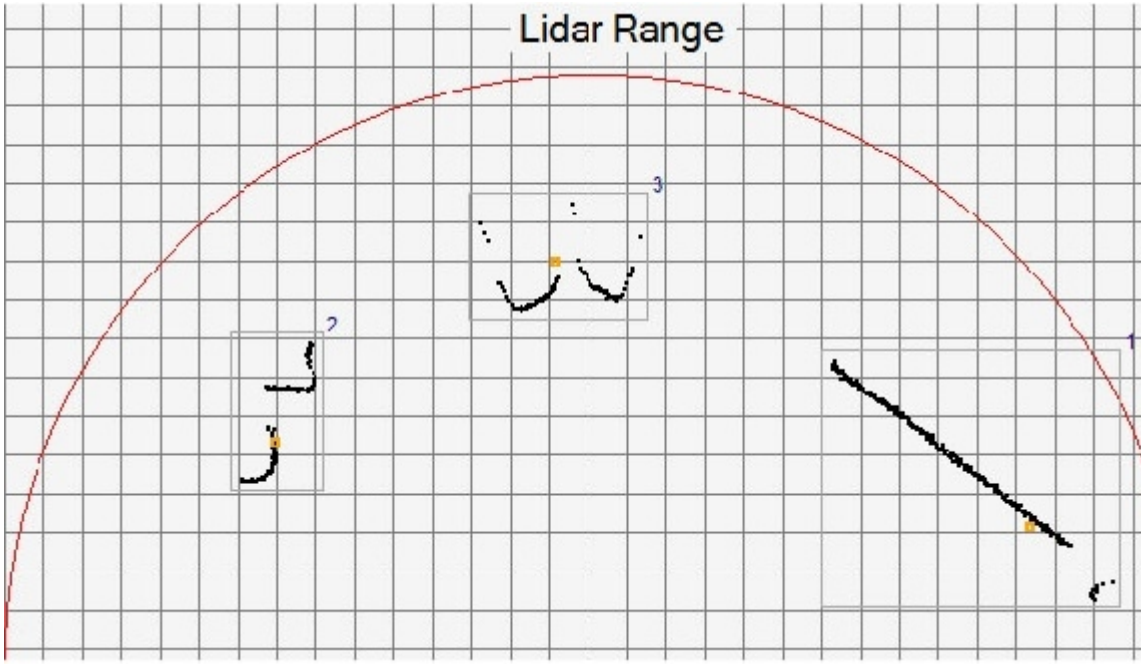

Fig. 4.10. Clustered view of two moving humans

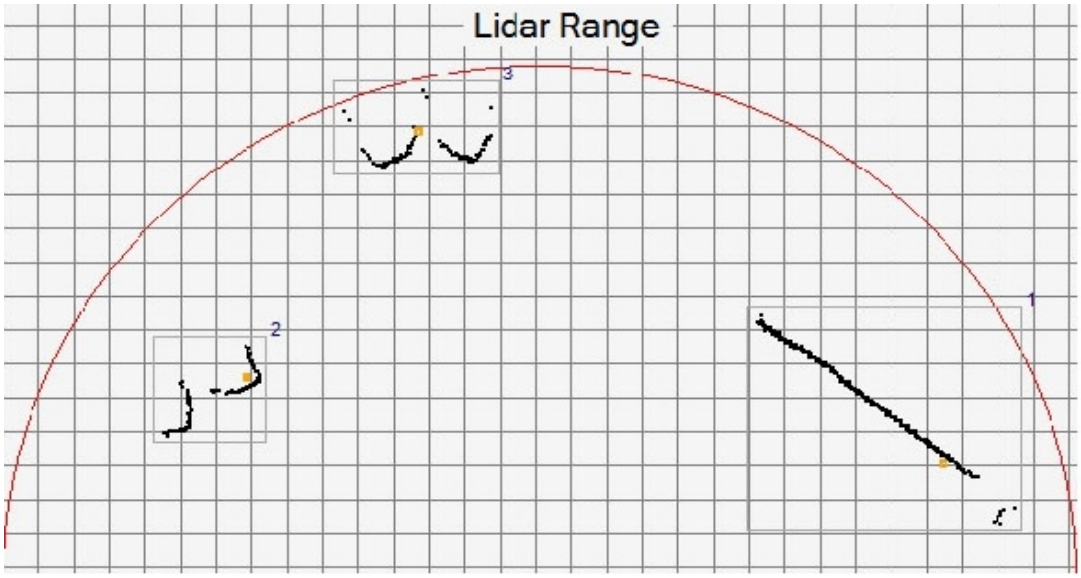

Fig. 4.11. Clustered view of two moving humans after 2 seconds from screenshot time of Figure 4.10.

4.2.5 . Test with Fixed LIDAR Position + Dynamic Clusters (Car)

Another test is made with fixed LIDAR and a moving car as well as an immobile car in a dynamic environment. First test is with a single walking human while the second is with two walking humans. Moving car has a dynamic speed between 0-10km/h. Environemental variables are given below:

Min range: 22mm,
Max range: 40000mm,
Minimum cluster length: 1000mm,
Max length between 2 points in a cluster: 500mm,
Max displacement: 2000mm/sec

Figures 4.12, 4.13, 4.14 and 4.15 show LIDAR detection output as well as filtered clusters. The tracking of a moving car is demonstrated by the help of camera images and can be distinguished with the unique ID 470 from LIDAR outputs.

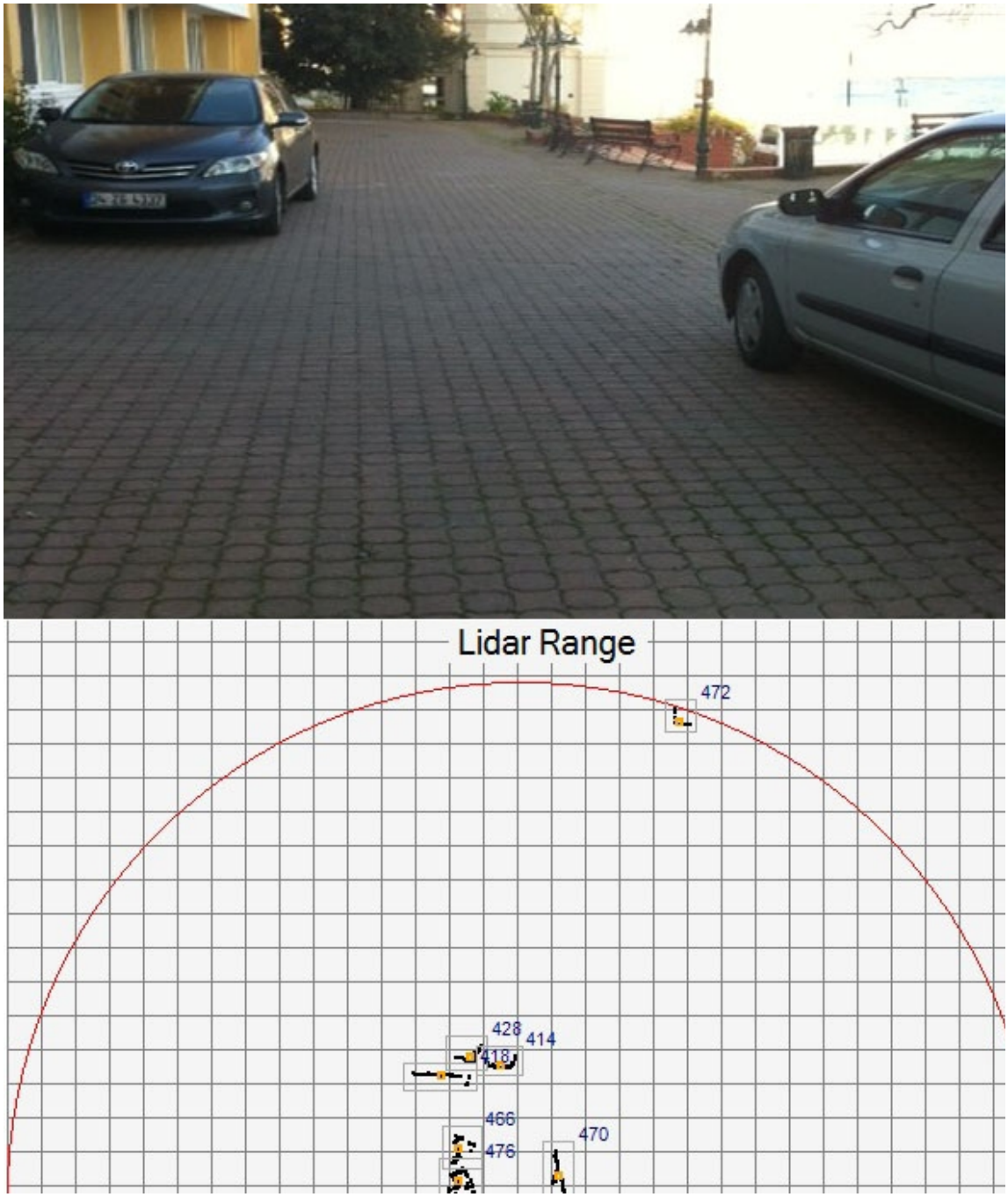

Fig. 4.12. Shows detection and tracking of test scene 1.

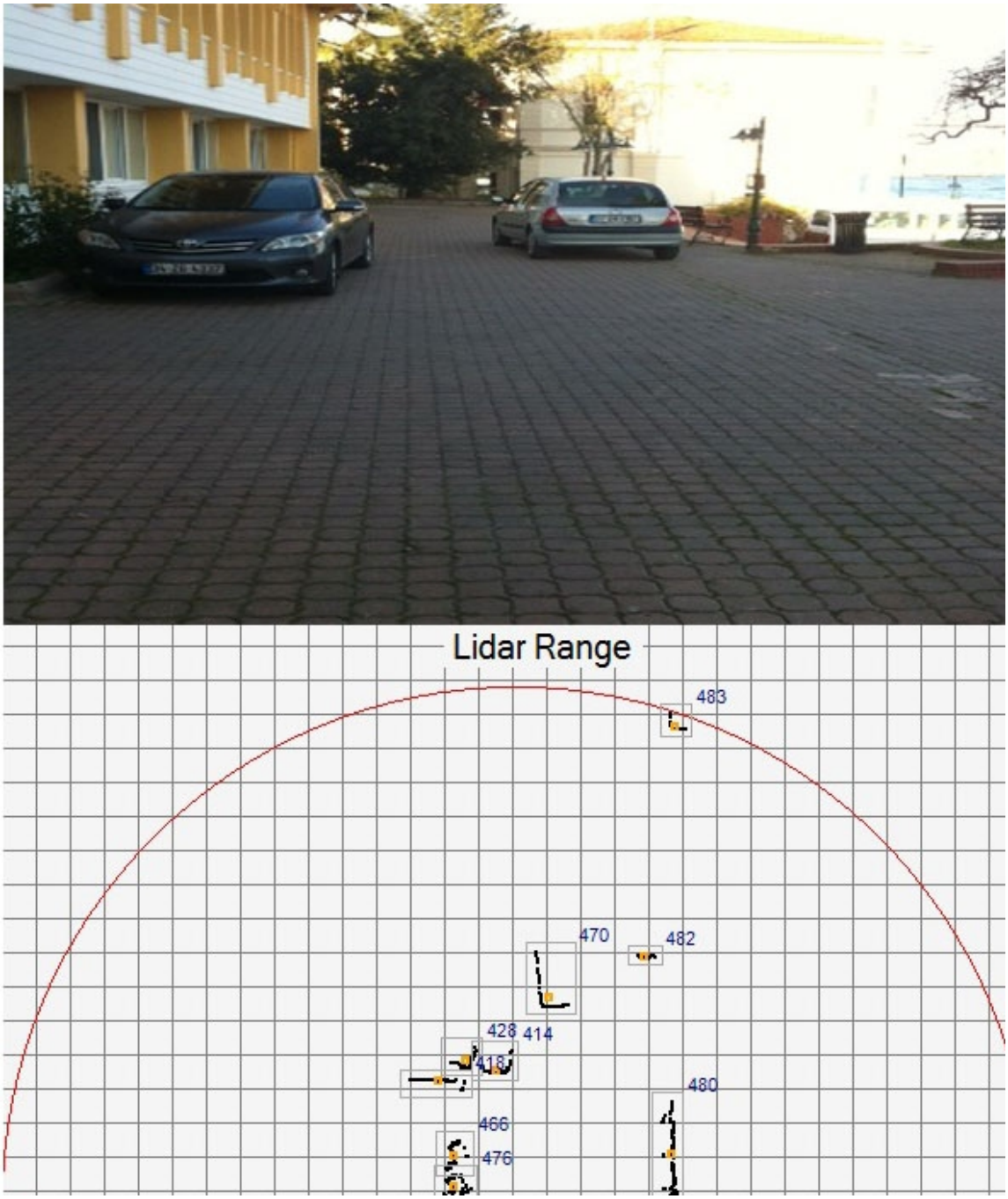

Fig. 4.13. Shows detection and tracking of test scene 2.

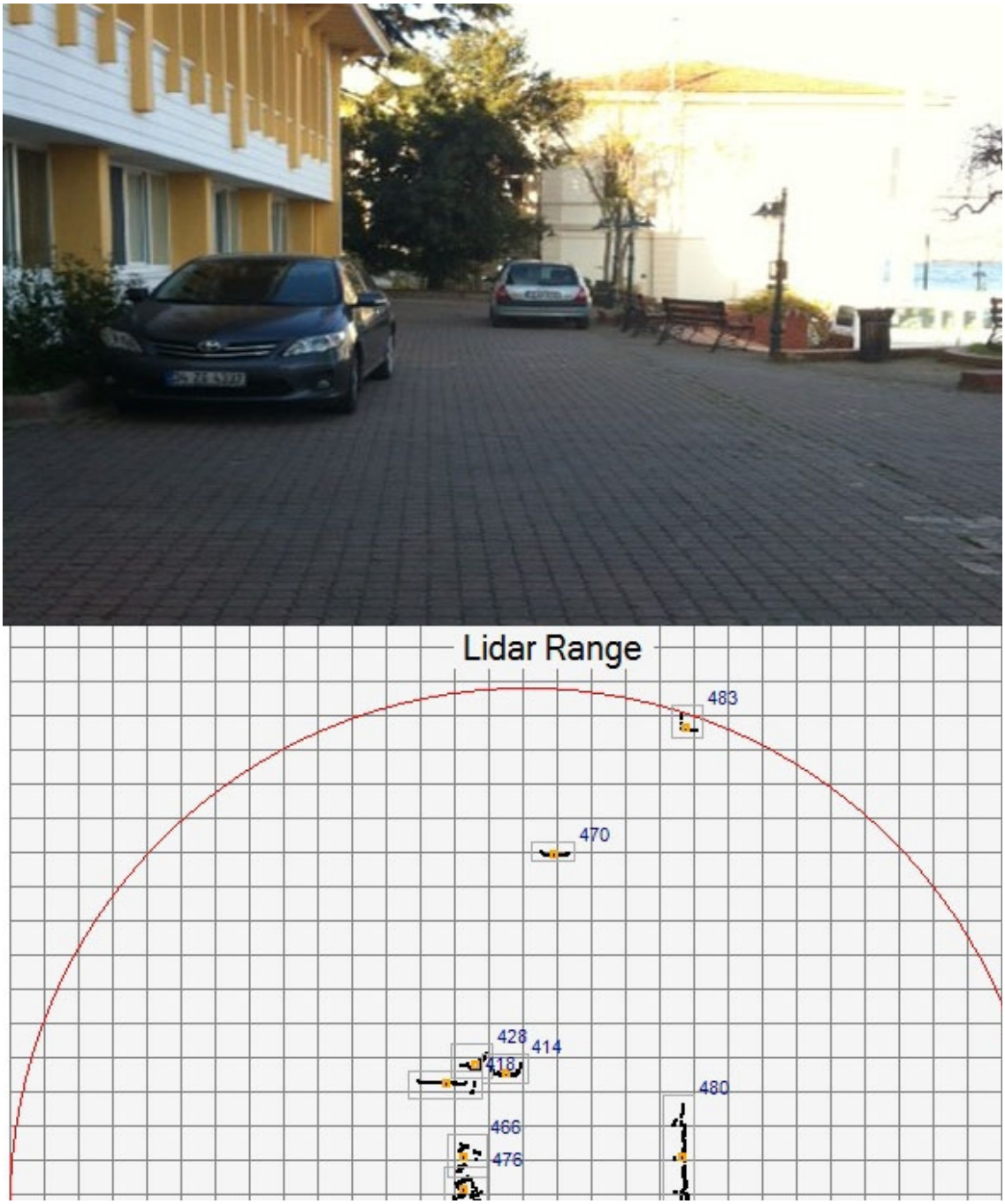

Fig. 4.14. Shows detection and tracking of test scene 3.

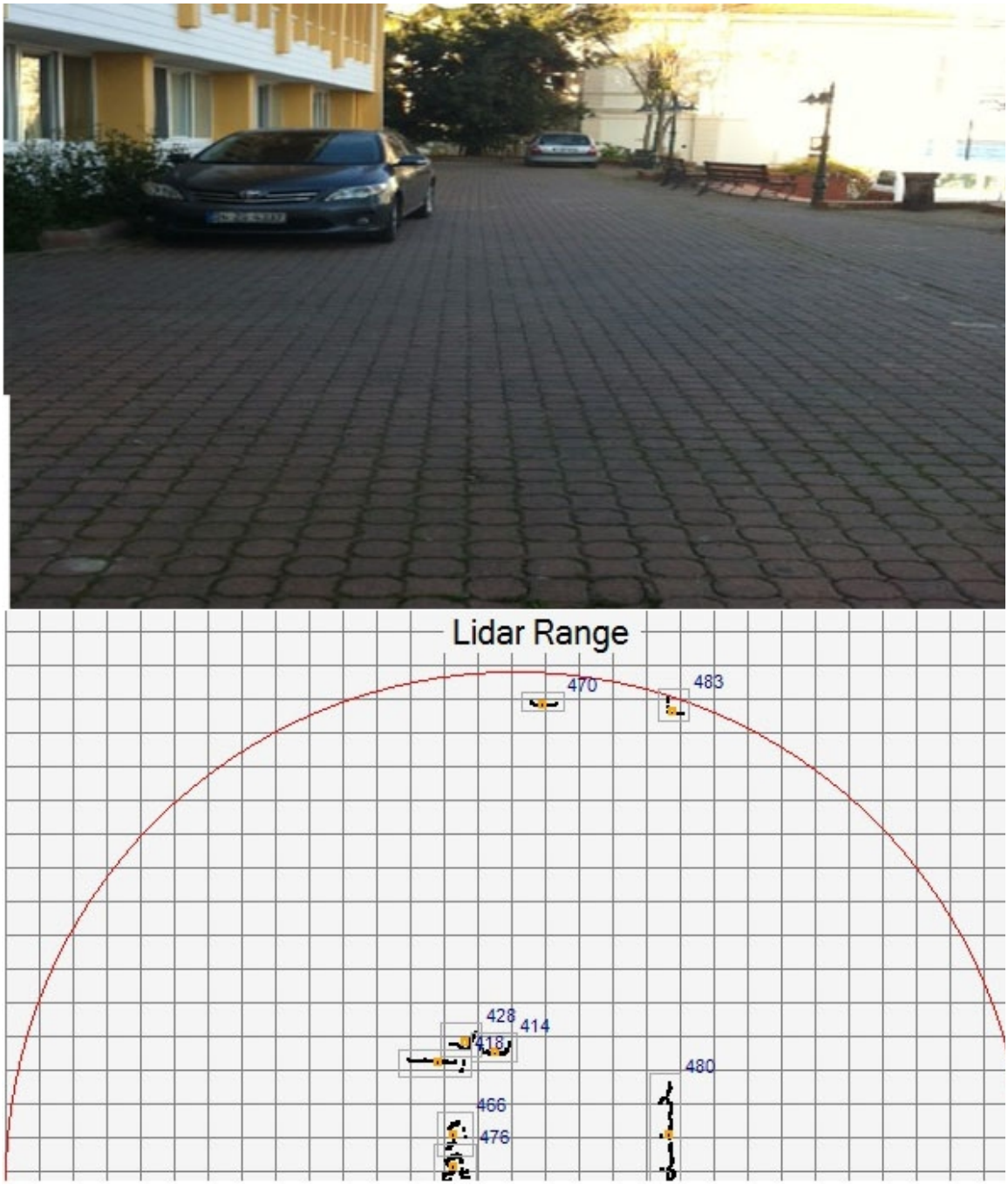

Fig. 4.15. Shows detection and tracking of test scene 4.

4.2.6 . Second Test with Fixed LIDAR Position + Dynamic Clusters (Car)

Another test is made with fixed LIDAR and a moving car as well as an immobile car in a dynamic environment. Moving car has a dynamic speed between 0-30km/h. Environemental variables are given below:

Min range: 22mm,

Max range: 40000mm,

Minimum cluster length: 800mm,

Max length between 2 points in a cluster: 500mm,

Max displacement: 2000mm/sec

Figures between 4.16 – 4.23 show LIDAR detection output as well as filtered clusters. The tracking of a moving car is demonstrated by the help of camera images and can be distinguished with the unique ID 11 from LIDAR outputs.

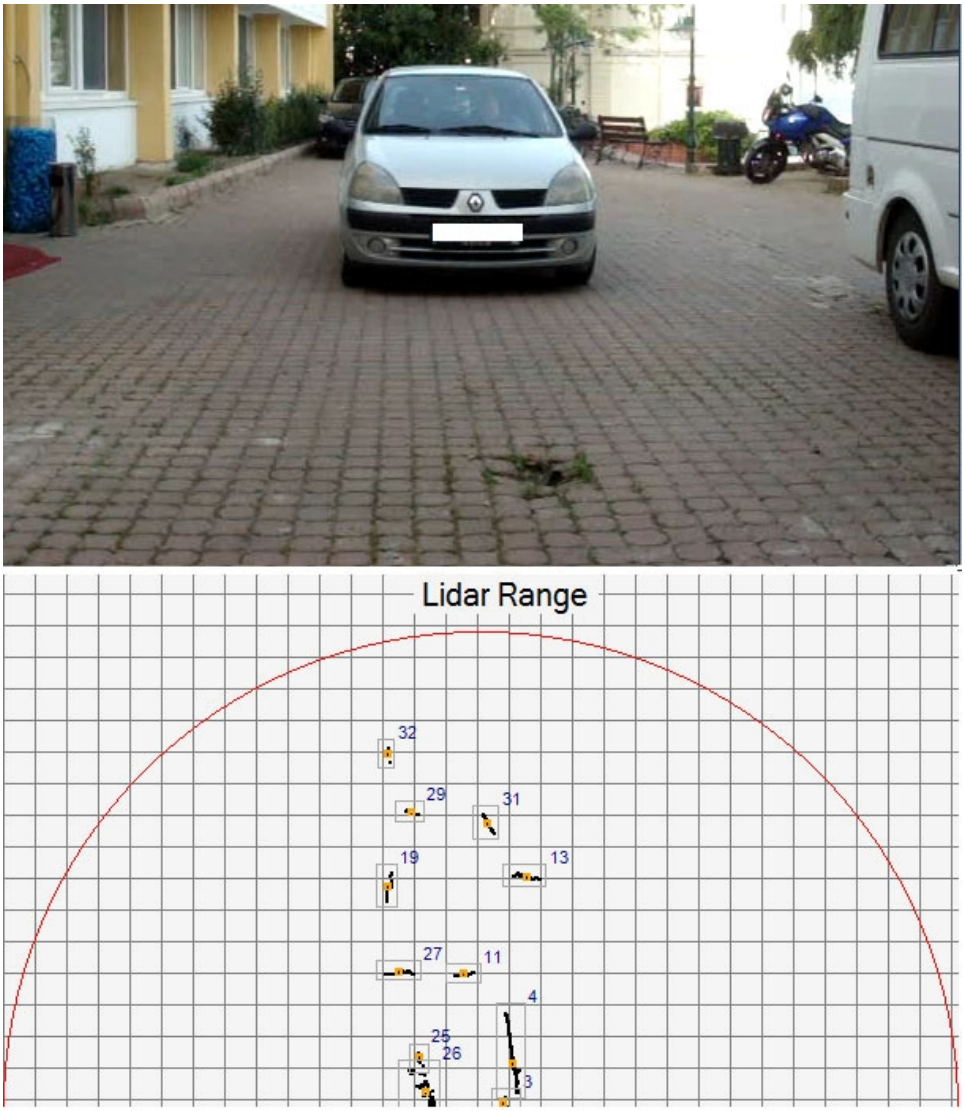

Fig. 4.16. Shows detection and tracking of test 2 scene 1.

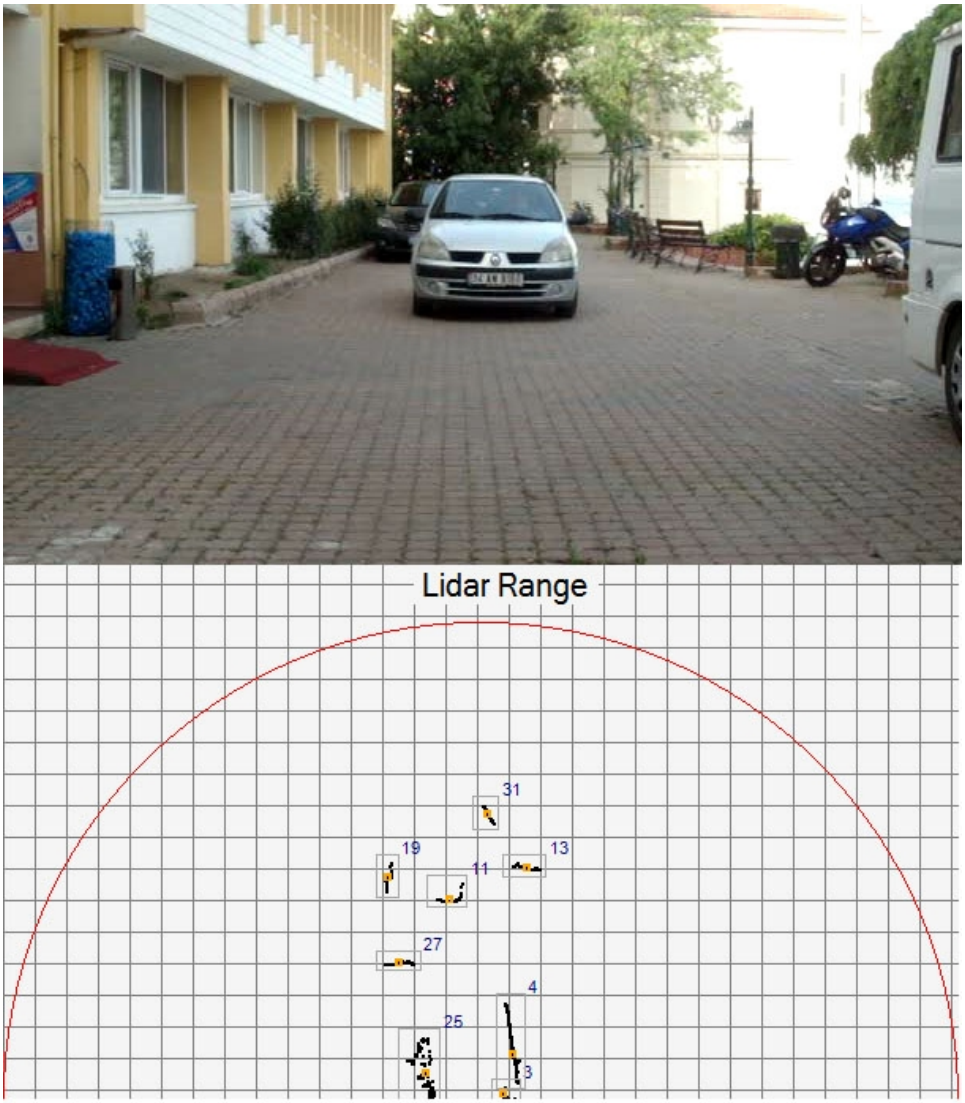

Fig. 4.17. Shows occlusion between cars with ID 29 and 11 of test 2 scene 2.

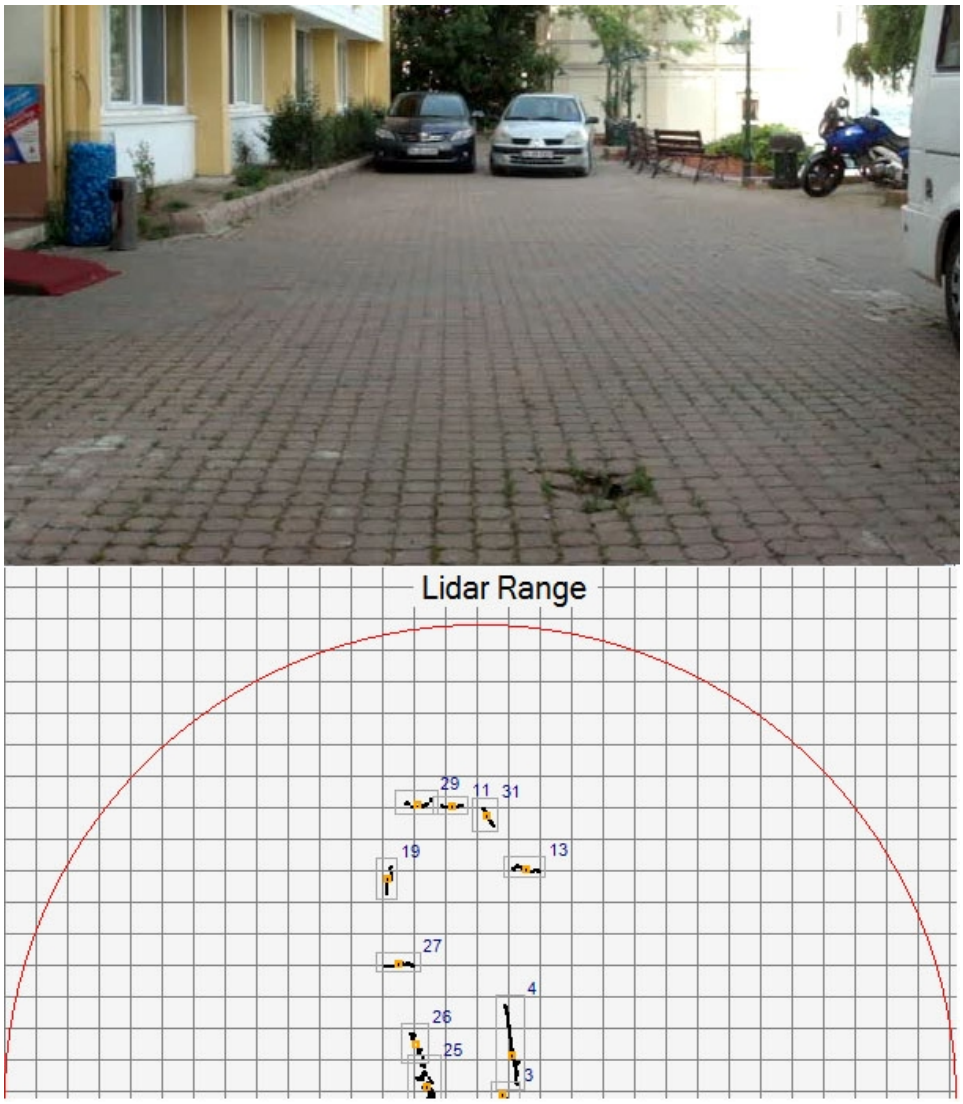

Fig. 4.18. Shows occlusion between cars with ID 29 and 11 of test 2 scene 3.

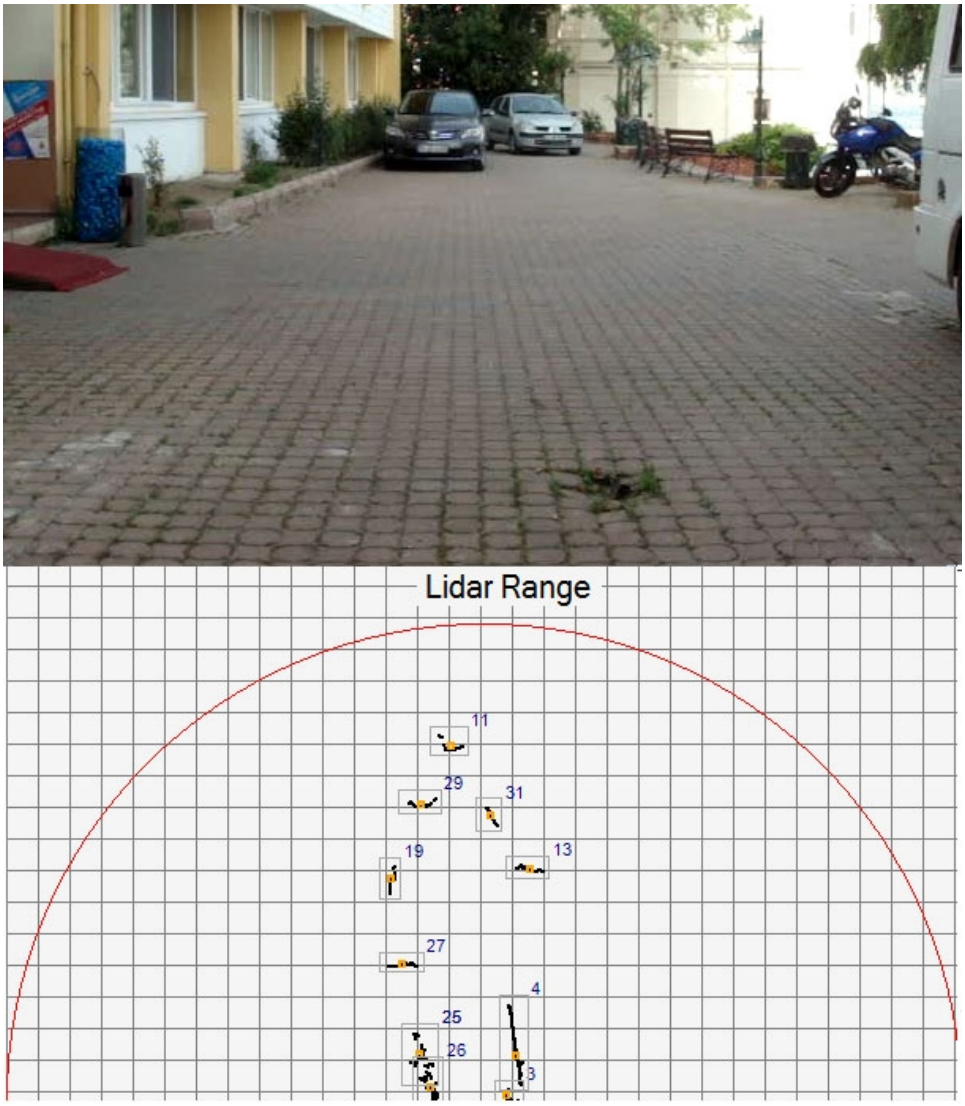

Fig. 4.19. Shows occlusion between cars with ID 29 and 11 of test 2 scene 4.

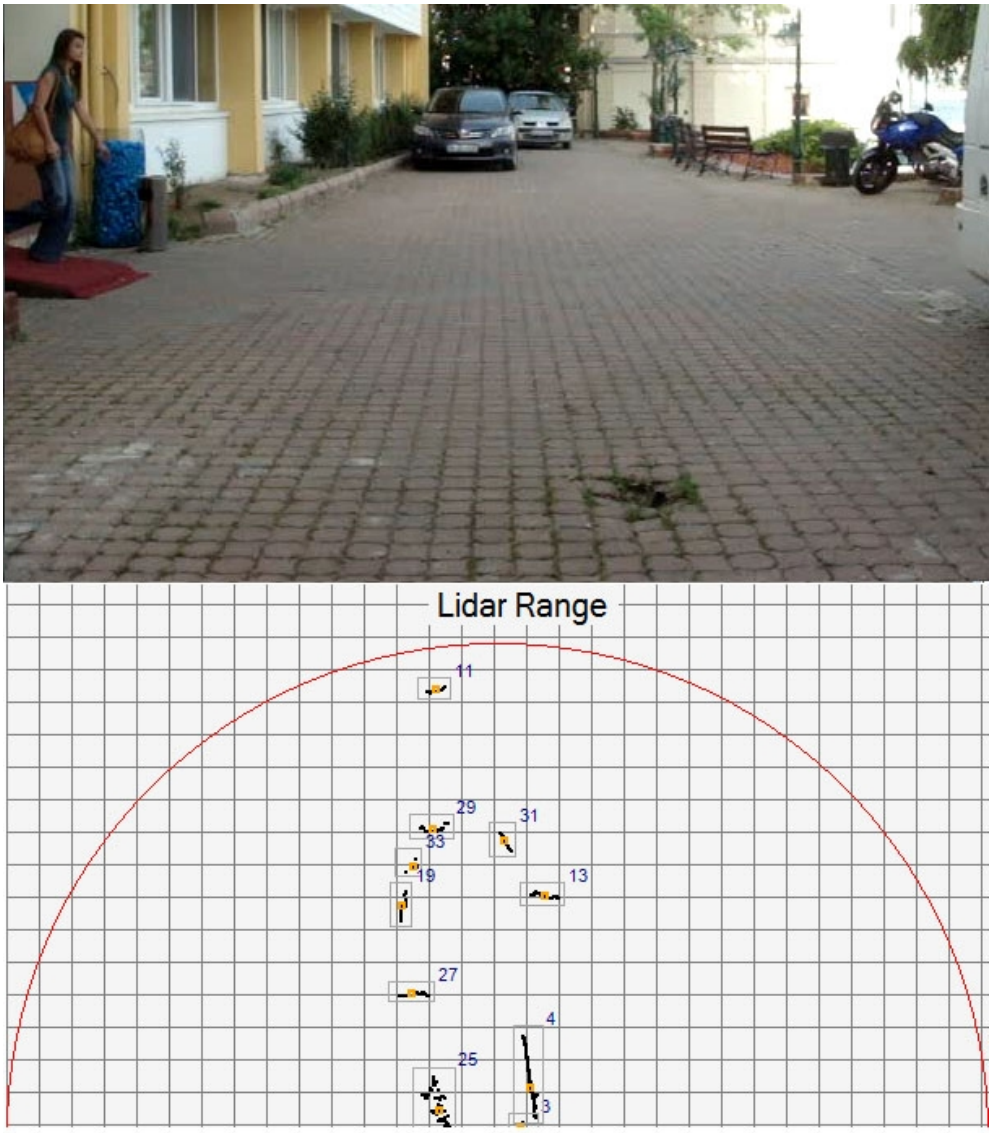

Fig. 4.20. Shows occlusion between cars with ID 29 and 11 of test 2 scene 5.

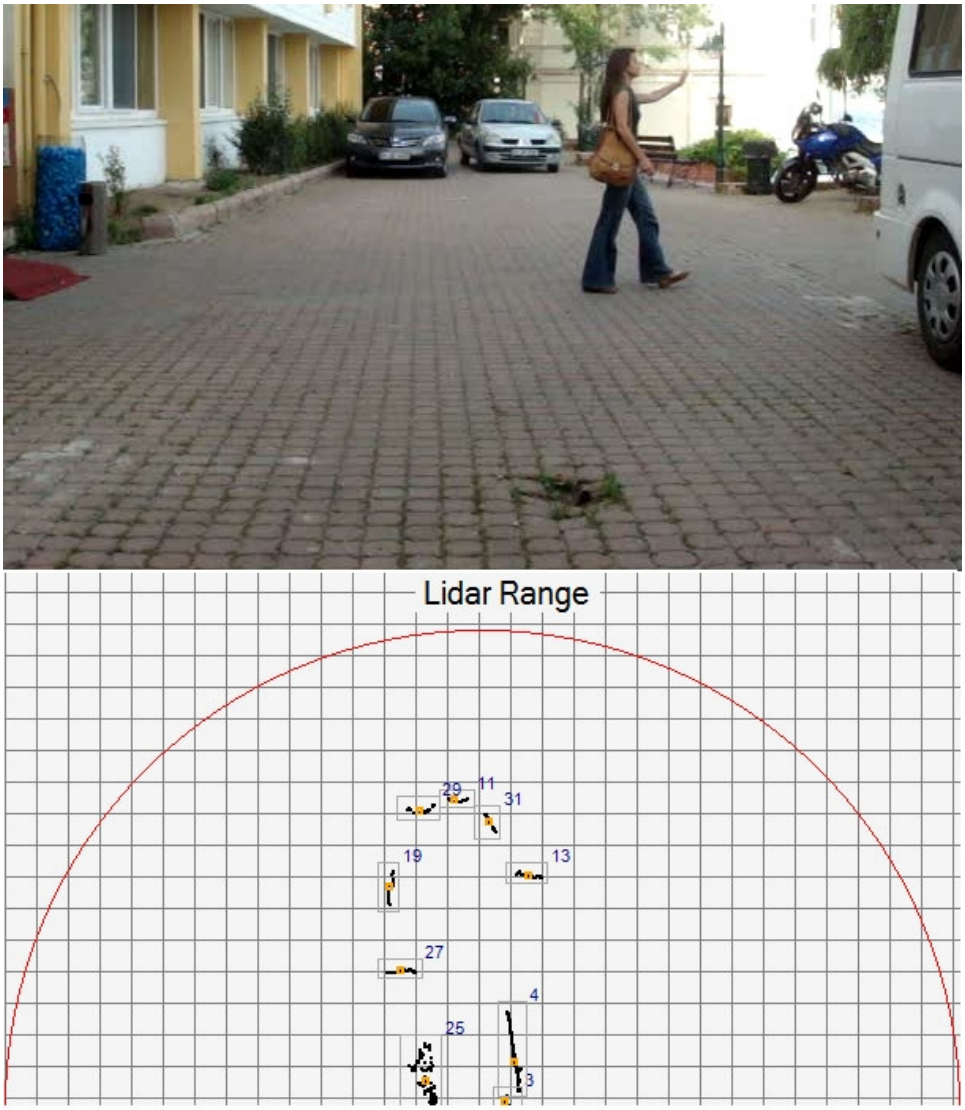

Fig. 4.21. Shows occlusion between cars with ID 29 and 11 of test 2 scene 6.

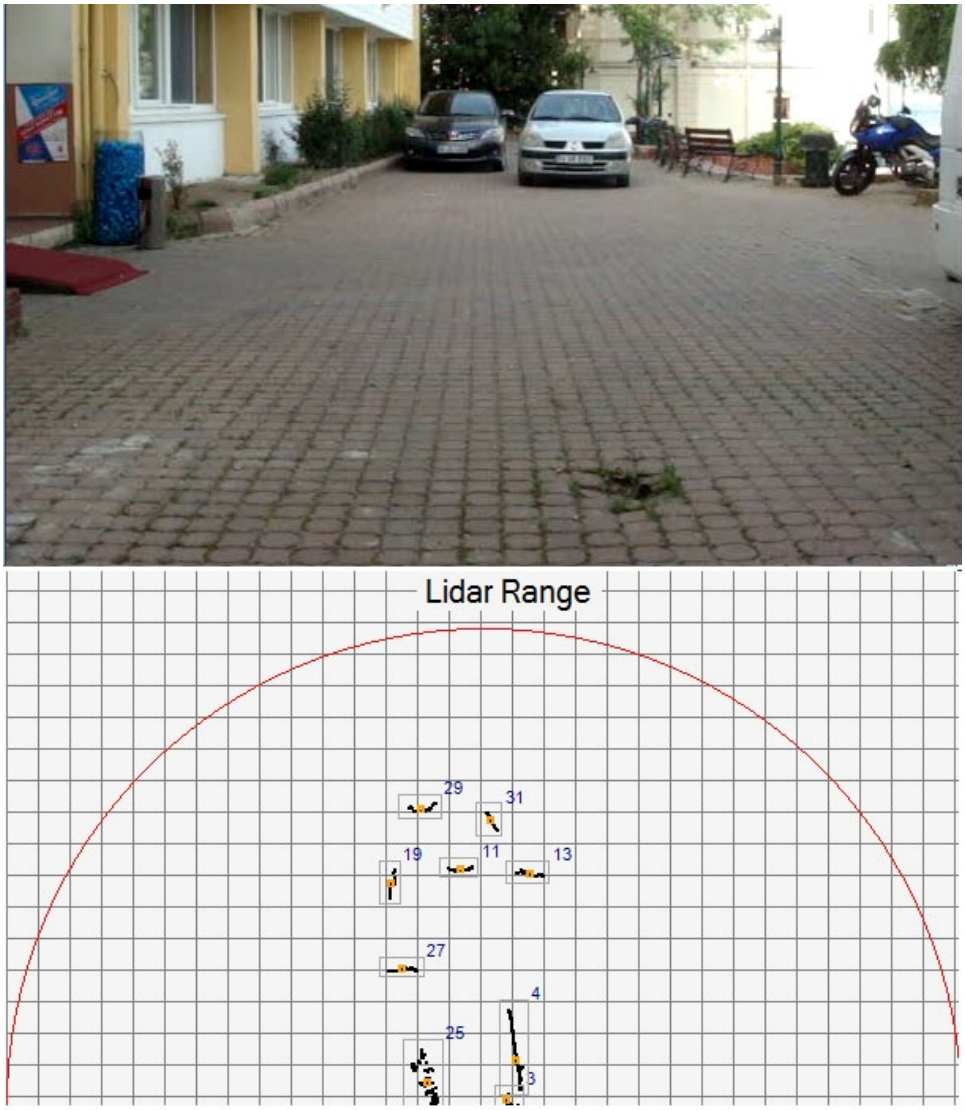

Fig. 4.22. Shows occlusion between cars with ID 29 and 11 of test 2 scene 7.

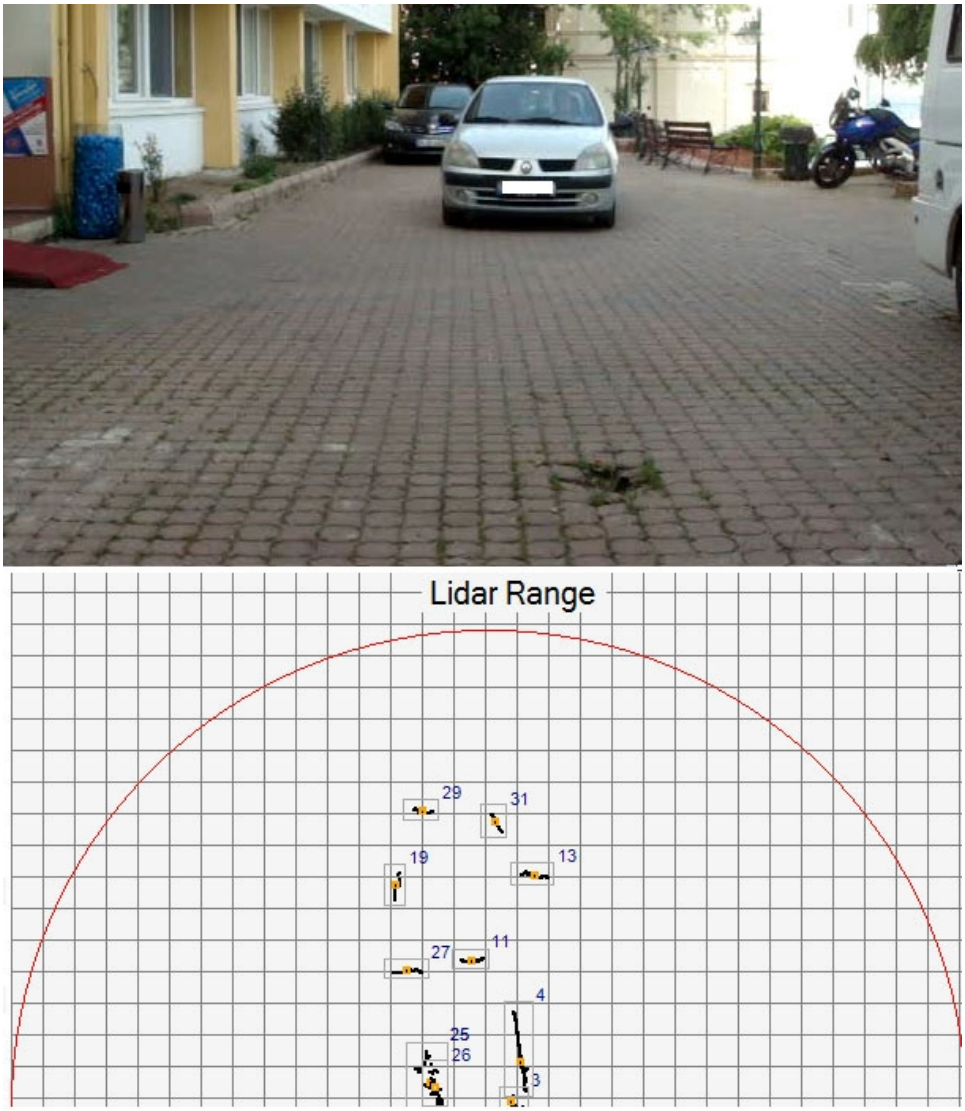

Fig. 4.23. Shows occlusion between cars with ID 29 and 11 of test 2 scene 8.

4.2.7 . Test with IMU, Odometer, GPS and Particle Filter

As the use of IMU is necessary for tracking clusters with dynamic LIDAR position, several tests are done. Especially the use of Particle Filter combined with Odometer and GPS has noticeable effect when correcting IMU acceleration rates. As it can be seen in Fig. 4.24., there is a graph of dynamic LIDAR position, which is accelerated and then stopped suddenly. Although the velocity is zero after stopping, without the use of Particle Filter, Odometer and GPS, the data from IMU has lot of noise even though bias values are correctly calculated.

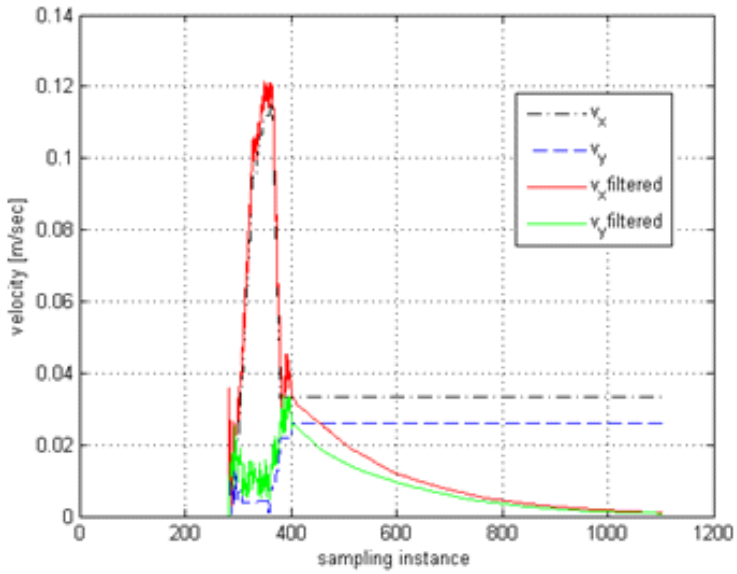

Fig. 4.24. Shows the Particle Filter effect to the velocity of a dynamic LIDAR.

5 . Application

5.1 . Overview

The application is written using C# using Microsoft Visual Studio 2010. Net Framework 3.5 must be installed in order to run the application. Basically, the application consists of four principal stages. It is designed to be multi-threaded and non-blocking. A different thread is used for each of the principal stages and the GUI thread. The principal stages are as follows:

- LIDAR
- IMU
- ODOMETER
- GPS

5.2 . Detailed Description

5.2.1 . LIDAR Communication

LIDAR connection stage is used to establish communication with LIDAR scanner. As LIDAR scanner in our implementation communicates via the ethernet port, IP and Port of the device must be set before establishing the communication. Data flow starts once the Start Capture is clicked. Parameters can be set from the control panel on the left hand side. On the right side, the effects of objects, clusters and Particle Filter on the clusters are shown in the LIDAR range. The main parameters that can be set are :

1. Minimum LIDAR range in mm
2. Maximum LIDAR range in mm
3. Minimum Cluster length in mm which is a calculated distance of two most distant points of Cluster.

4. Maximum length between two points in a cluster in mm
5. Maximum deplacement of Cluster centroid in mm/sec
6. Whether to show only Cluster or not
7. Whether to combine Clusters of which the centroids are closer than the maximum length between two points in a cluster parameter.
8. Whether to filter clusters by the minimum cluster length or not
9. Whether to show Particle Filter's effect on the clusters or not
10. Whether to show historical Particle Filter effect data or not
11. Whether to show historical centroid position data or not

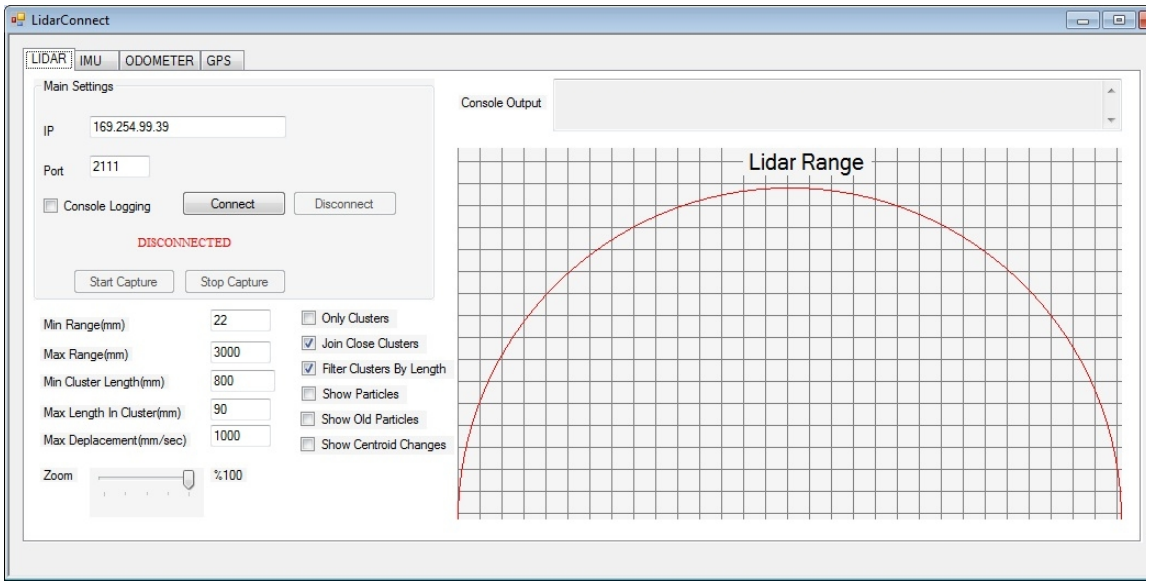

Fig. 5.1. Shows the LIDAR control window of the software

5.2.2 . IMU Communication

IMU connection stage is used to communicate with the IMU. After chosing the corresponding COM port, the connection to the device occurs automatically. Data flow starts once the Start Capture is clicked. Before carrying out any calculations, first, the application attempts to compute bias values which are then used for more accurate acceleration, yaw rate and speed calculations. X/Y/Z acceleration rates and X/Y/Z yaw rates are received from the IMU. X/Y/Z speed rates are calculated by using acceleration and time values.

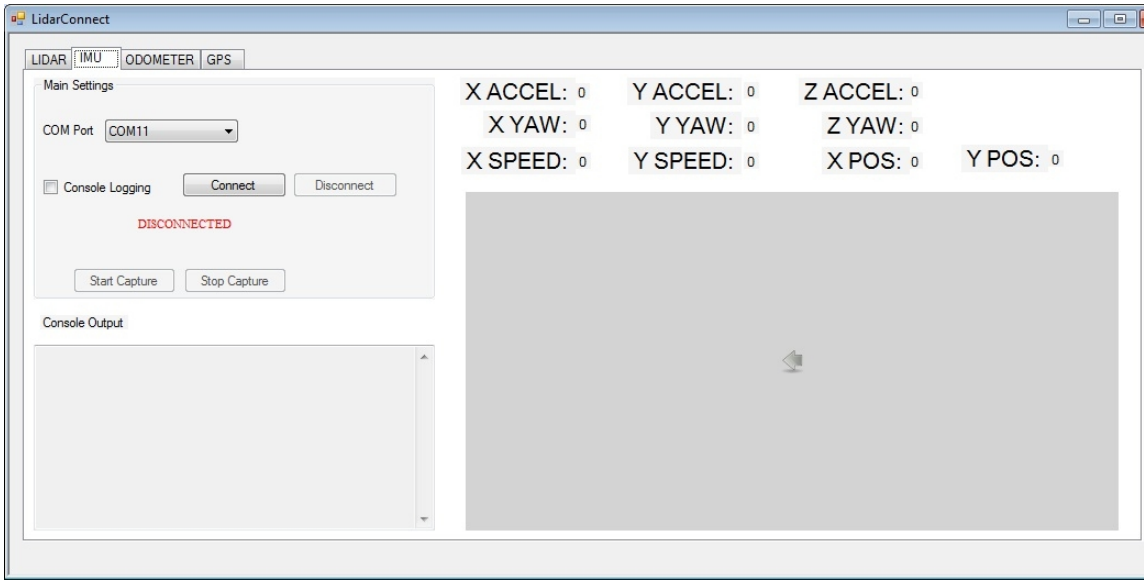

Fig. 5.2. Shows the IMU control window of the software

5.2.3 . Odometer Communication

ODOMETER connection stage is used to communicate with the Roboteq Odometer and Motor Controllers. The connection is made after chosing the related COM port. Data flow starts after clicking the Start Capture button. In the command control panel, robot movements can be controlled. On the right-hand side, the information shown includes left/right motor rotations, left/right motor speeds, total distance covered and robot general status.

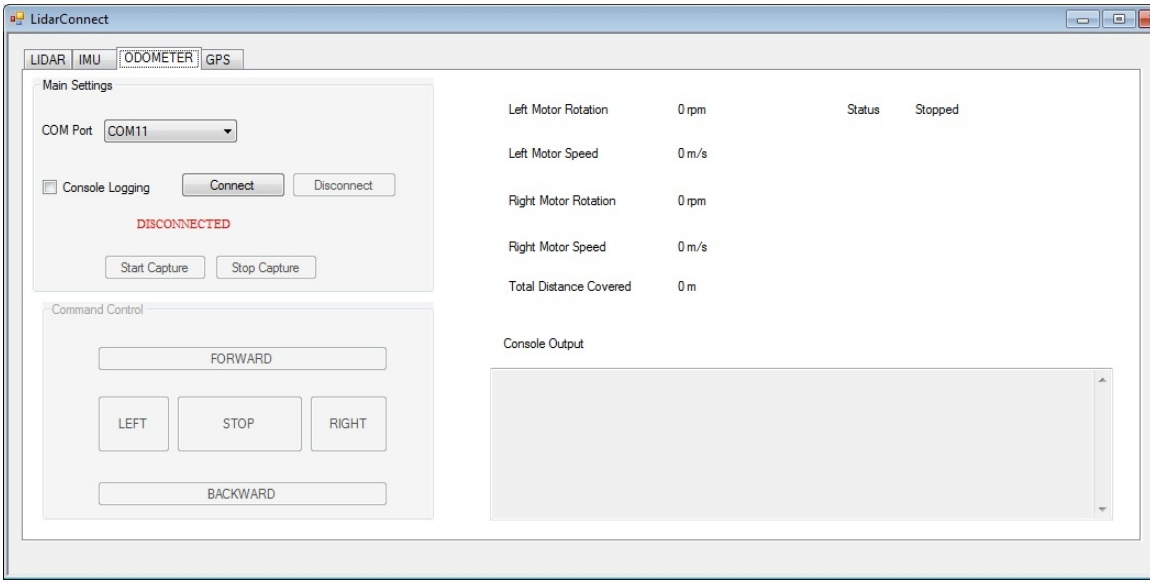

Fig. 5.3. Shows the Odometer control window of the software

5.2.4 . GPS Communication

GPS connection stage is used to communicate with the GPS device. After the COM port of the device is selected, data flow begins automatically. Location, antenna altitude, Geoidal separation, Velocity and communicated satellites are shown on the right of the information panel.

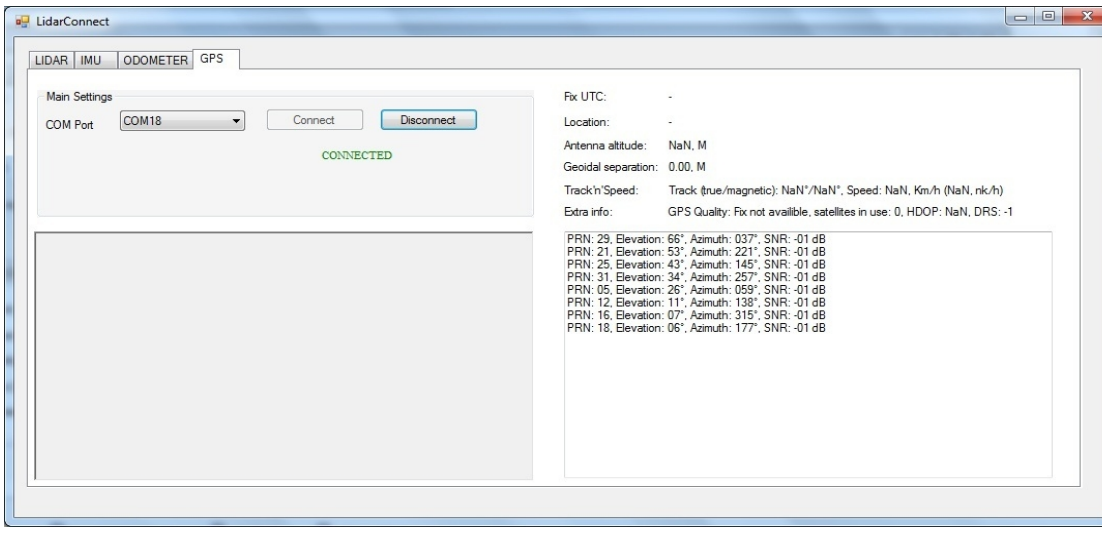

Fig. 5.4. Shows the GPS control window of the software

6 . Performance Analysis

Software performance tests are done in 3 stages including CPU sampling by monitoring CPU usage of the software as well as usage of different cores, memory allocation by tracking managed memory allocation and concurrency by detecting threads which are waiting for other threads. The tests are applied in a computer with the following hardware specifications:

- Intel Core 2 Duo CPU 2.8 GHz
- 4 GB DDR2 Ram

Visual Studio 2010 analyzer tool is used for performing tests.

6.1 . CPU Sampling

CPU sampling tests are done especially to detect CPU usage with and without Particle Filter. Methods that are most expensive for the CPU are also analysed to reduce the time spent by optimizing them. Fig.6.1 and Fig.6.2 show the CPU usage of the software with and without Particle Filter. It can be inferred that the average CPU usage with Particle Filter is greater than the average CPU usage without Particle Filter.

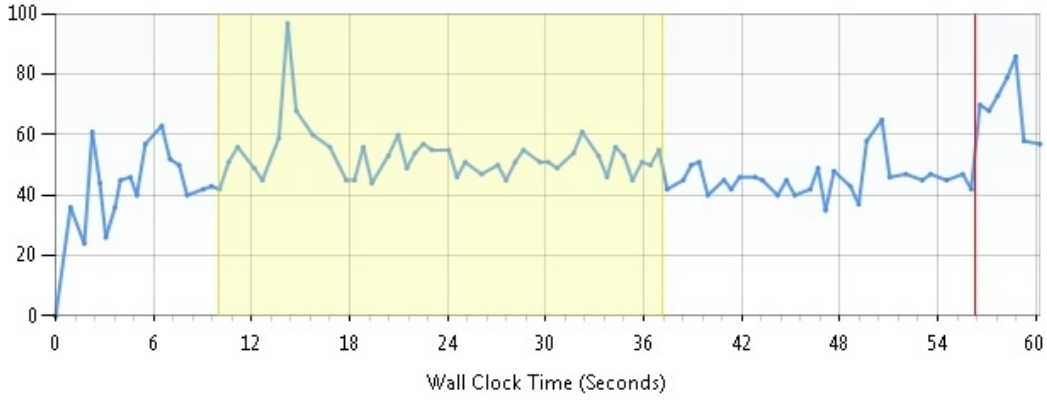

Fig. 6.1. Shows CPU usage of the software without Particle Filter

Besides, %48 of the total CPU usage without Particle Filter is occupied by a function that gets points from Lidar and then match with the previous points in order to identify each cluster. This is the most expensive operation of all methods. It contains methods to detect clusters, compare clusters and rotate/translate each point by receiving data from IMU. In fact, by analyzing the CPU usage of these sub methods, it can be clearly seen that the most expensive sub method is the cluster detection stage with %37 of the entire CPU usage. While the cluster comparison method takes %4.18, translation/rotation stage takes only %0.06. A considerable amount of time is also spent to receive data from LIDAR which is up to %4.38.

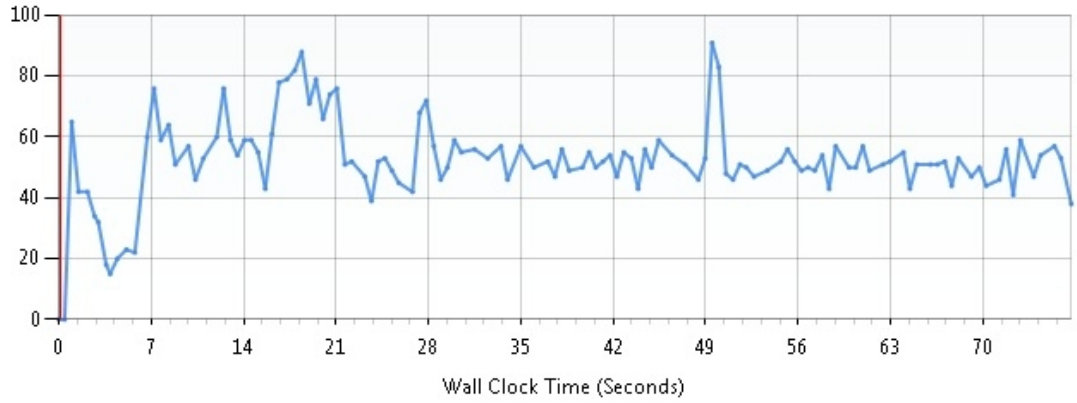

Fig. 6.2. Shows CPU usage of the software with Particle Filter

6.2 . Tracking Memory Allocation

Memory allocation tracking is done in two consecutive tests to compute the average memory allocation of the software and also to compare memory allocations with and without Particle Filter. After 33 seconds of usage, 182,925,776 total bytes are allocated by the software while Particle Filter is not in use. However, this allocation amount is increased to 225,471,309 total bytes with Particle Filter. When analyzing functions that allocate most memory, the first place is taken by a generic list which stores cluster histories with %25.21 of the total bytes used. Furthermore binary reading operations take %18.73 of the allocated memory and can be reduced with optimization. Memory allocation of Particle Filter algorithm is up to %7.50 when is in use.

6.3 . Concurrency

It is possible that the computer may be simultaneously serving thousands of clients with thousands of processes in execution. In such a scenario, it is possible that the CPU may enter into a dead-lock situation. Dead-lock is a situation on a computer when one process (say A) locks a dedicated resource (say X) and waits for another dedicated resource (say Y) locked by another process (say B) which is looking for the resource

(X) locked by the first process (A) rendering both the processes in an infinite wait for the non-available resource. Handling such situations is referred to as concurrency control in software development circles. There are many methods for handling concurrency but these situations cannot be detected by normal functional testing. This issue needs to be detected using the concurrent testing.

Analyzing the concurrency tests, no dead-lock is detected as the software is coded entirely to be non-blocking. Average CPU utilization by the software is %30 of the entire CPU. %79 of this CPU time is spent in synchronization while Execution takes %6 and UI Processing %5. In Fig.6.5., it can be clearly seen that there is not any blocking thread that comes forward. Despite disadvantages of a blocking software, a blocking design could have also been adopted for that project due to the fact that much more effort needs to be shown during data capture from different hardware devices such as LIDAR and IMU which generally are working with different frequencies. In fact, to synchronize them easily and with a less effort, a blocking design could be developed, although, the results of concurrency test would not be as good as non-blocking case.

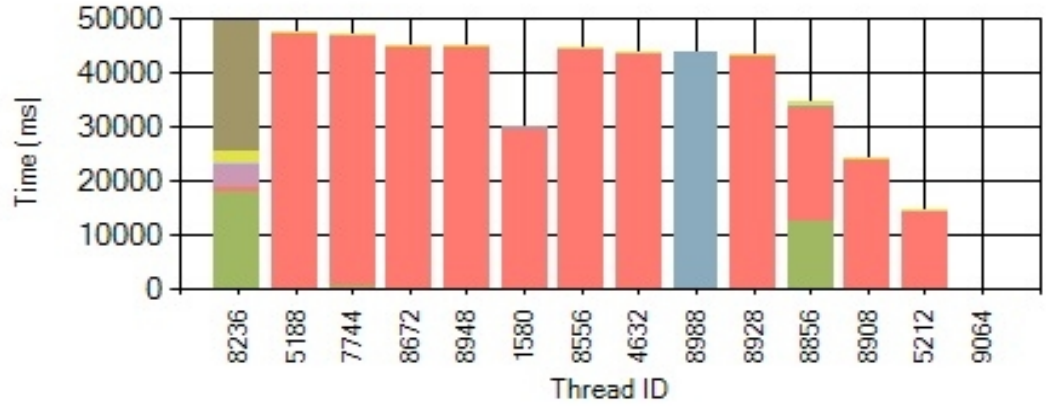

Fig. 6.3. Shows per-thread summary

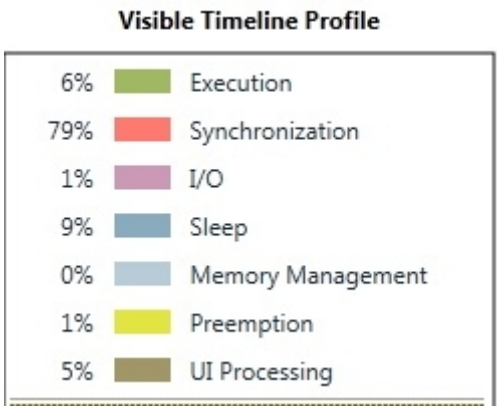

Fig. 6.4. Shows the timeline profile

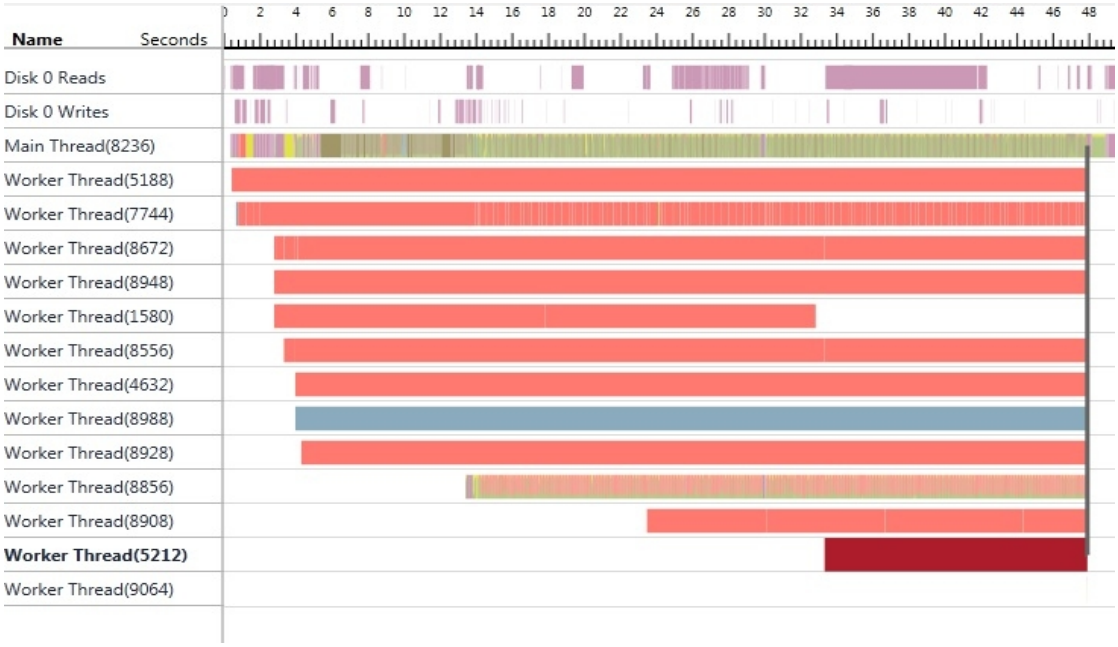

Fig. 6.5. Shows timeline for different worker threads as well as disk read/write.

7 . Conclusion

This thesis describes a distinct approach and implementation for dynamic object detection and tracking. The key idea is to predict a cluster's possible trajectories, taking into account the relative position/rotation changes of the LIDAR scanner, utilizing the Particle Filter Algorithm. We have the ability to detect and track objects in a specified range, especially with a good accuracy in a large environment, where clusters do not collide with each other and do not fall into the same angle at the same time. Furthermore, objects that are needed to be tracked must be known in advance (whether they are cars, humans, etc.), since their maximum displacement in a given time interval should be calculated and their approximate shape must be estimated, in order to accurately predict their possible trajectories. In summary, multiple dynamic clusters allow us to detect and track objects despite the presence of changing ranges and different object types. Future work can be mainly focused on:

- Extracting the algorithm with a specific detection system for different object types and implementing an edge detection system.
- Developing a new algorithm which will analyze the cluster's position changes and help calculating the position prediction of each cluster.
- Implementing more LIDARs to extract the view
- Integrating a camera which will help increasing the detection accuracy in fusion with LIDAR data

References

Arras, K.O., Castellanos, J.A., Schilt, M., Siegwart, R. (2002). Feature-based multi-hypothesis localization and tracking for mobile robots using geometric constraints. In Procedings of IEEE International Conference on Robotics and Automation (ICRA), 2(1), p.1371-1377.

Betke, M., Haritaoglu, E., Davis, L.S. (2000). Real-time multiple vehicle detection and tracking from a moving vehicle. In: International Conference on Machine Vision and Applications, p.69-83.

Blanc, C., Trassoudaine, L., Gallice, J. (2005). EKF and Particle Filter Track to Track Fusion : a quantitative comparison from Radar/LIDAR obstacle tracks. In: 7th International Conference on Information Fusion, 2(1).

Broggi, A., Bertozzi, M., Fascioli, A., Sechi, M. (2000). Shape-based pedestrian detection. In Proceedings of IEEE Intelligent Vehicles Symposium, p.215-220.

Cho, J.U., Jin, S.H., Pham, X.D., Jeon, J.W. (2006). Object Tracking Circuit using Particle Filter with Multiple Features. In: SICE-ICASE International Joint Conference, p.1431-1436.

Doucet, A., Vo, B.N., Andrieu, C., Davy, M. (2002). Particle filtering for multi-target tracking and sensor management. In: Proceedings of 5th International Conference on Information Fusion, 1(1), p.474-481.

Durrant-Whyte, H. (2006). Multi Sensor Data Fusion. Australian Center for Field Robotics, The University of Sydney NSW, 1(2), Australia

Ess, A., Leibe, B., Schindler, K., Gool, L.V. (2009). Moving obstacle detection in highly

dynamic scenes. In: International Conference on Robotics and Automation (ICRA), p.56-63.

Ess, A., Schindler, K., Leibe, B., Gool, L.V. (2010). Object detection and tracking for autonomous navigation in dynamic environments. In: Internation Journal of Robotics Research, 29(14), p.1707-1725.

Fortin, B., Noyer, J.C., Lherbier, R. (2012). A particle filtering approach for joint vehicular detection and tracking in lidar data. In: Instrumentation and Measurement Technology Conference (I2MTC), p.391-396.

Franke, U., Heinrich, S. (2002). Fast obstacle detection for urban traffic situations. In: IEEE Trans. Intell. Transport. Syst., 3(3), p.173-181.

Gao, B., Coifman, B. (2006). Vehicle Identification and GPS error detection from a Lidar Equipped Probe. In: Intelligent Transportation Systems Conference (ITSC), p.1537-1542.

Hancock, J. (1999). Laser intensity-based obstacle detection and tracking. Ph.D. Thesis. The robotics institute, Carnegie Mellon University, Pittsburgh, USA.

Lee, K.W., Kalyan, B., Wijesoma, S., Adams, M., Hover, F.S., Patrikalakis, N.M. (2010). Tracking Random Finite Objects using 3D-LIDAR in Marine Environments. In: Proceedings of ACM Symposium on Applied Computing, p.1282-1287.

Leibe, B., Schindler, K., Gool, L.V. (2007). Coupled detection and trajectory estimation for multi-object tracking. In: Internation Conference on Computer Vision (ICCV), p.1-8.

Murshed, M., Kabir, H., Chae, O. (2011). Moving object tracking - An edge segment based approach. International Journal of Innovative Computing, Information and Control, 7(7), p.3963-3979.

Neira, J., Tardos, J.D., Horn, J., Schmidt, G. (1999). Fusing range and intensity images for mobile robot localization. In: IEEE Trans. Robotics and Automation", 15(1), p.76-84.

Oren, M., Papageorgiou, C., Sinha, P. (1997). Pedestrian detection using wavelet templates. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p.193-199.

Ozguner, U., Acarman, T., Redmill, K. (2011). Autonomous Ground Vehicles. Artech House Intelligence.

Premebida, C., Monteiro, G., Nunes, U., Peixoto, P. (2007). A Lidar and Vision-based Approach for Pedestrians and Vehicle Detection and Tracking. In: Intelligent Transportation Systems Conference (ITSC), p.1044-1049.

Schulz, D., Burgard, W., Fox, D., Cremers, A.B. (2001). Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In: Proceedings of Robotics and Automation (ICRA), 2(1), p.1665-1670.

Shan, C., Tan, T., Wei, Y. (2007). Real-time hand tracking using a mean shift embedded particle filter". Journal Elsevier Science Inc , 40(7), p.1958-1970.

Sotelo, M.A., Parra, I., Fernandez, D., Naranjo, E. (2006). Pedestrian detection using SVM and multi-feature combination. In: Intelligent Transportation Systems Conference (ITSC), p.103-108.

Tamas, L., Lazea, G. (2010). Pattern Recognition and Tracking Dynamic Objects with LIDAR. In: Robotics (ISR), p.1-6.

Toth, D., Aach, T. (2003). Detection and recognition of moving objects using statistical motion detection and Fourier descriptors. In: Proceedings of 12th Conference on Image Analysis and Processing (ICIAP), p.430-435.

Weigel, H., Lindner, P., Wanielik, G. (2009). Vehicle Tracking with Lane Assignment by Camera and LIDAR Sensor Fusion". In: Intelligent Vehicles Symposium, p.513-520.

Yu, J.X., Cai, Z.X., Duan, Z.H. (2008). Detection and tracking of moving object with a mobile robot using laser scanner. In: Machine Learning and Cybernetics, 4(1), p.1947-1952.

Yu, Q., Medioni, G., Cohen, I. (2007). Multiple target tracking using spatio-temporal Markov chain Monte Carlo data association. In: International Conference on Computer Vision and Pattern Recognition (ICCV) , p.1-8.

Biographical Sketch

Berk PELENK was born in İstanbul in March 4, 1986. He graduated from Notre Dame de Sion French High School in 2005. He received his B.S. Degree in Industrial Engineering in 2009 from Galatasaray University, İstanbul. He is currently a Master Of Science student in Computer Engineering in Institute of Science and Engineering at Galatasaray University, İstanbul.