Low Cost, Robust and Real Time System for Detecting and Tracking Moving Objects to Automate Cargo Handling in Port Terminals

Victor Vaquero[1], Ely Repiso[1], Alberto Sanfeliu[1],
John Vissers[2], and Maurice Kwakkernaat[2]

[1] Institut de Robotica i Informatica Industrial, CSIC-UPC
{vvaquero,erepiso,sanfeliu}@iri.upc.edu
[2] TNO, Helmond, The Netherlands
{john.vissers,maurice.kwakkernaat}@tno.nl

Abstract. The presented paper addresses the problem of detecting and tracking moving objects for autonomous cargo handling in port terminals using a perception system which input data is a single layer laser scanner. A computationally low cost and robust Detection and Tracking Moving Objects (DATMO) algorithm is presented to be used in autonomous guided vehicles and autonomous trucks for efficient transportation of cargo in ports. The method first detects moving objects and then tracks them, taking into account that in port terminals the structure of the environment is formed by containers and that the moving objects can be trucks, AGV, cars, straddle carriers and people among others. Two approaches of the DATMO system have been tested, the first one is oriented to detect moving obstacles and focused on tracking and filtering those detections; and the second one is focused on keepking targets when no detections are provided. The system has been evaluated with real data obtained in the CTT port terminal in Hengelo, the Netherlands. Both methods have been tested in the dataset with good results in tracking moving objects.

Keywords: Object detection, object tracking, DATMO, multi-hypothesis tracking, Autonomous driving, Autonomous transportation of cargo.

1 Introduction and State of The Art

In the actual globalized world the volume of international trade, of which up to 90% is fully containerized, keeps rising up. Cargo handling at port terminals is a key step across the international supply chain network [2], and there is an increase demand of automatizing systems.

In order to improve efficiency, increase capacity and speed up operations to meet future demands, port terminals requires the use of advanced technologies and automation. Changes in port operations have been gradually introduced, and this trend seems to continue, as it is shown that one of the least efficient and most costly processes in ports come from internal transportation in non-automated terminals [13]. Therefore, Automated Container Terminals (ACTs) with Automated Storage and Retrieval Systems - ASRS [11] or Automated Guided Vehicles - AGVs [13] are nowadays getting more presence around the world. In Europe, the European Combined Terminal (ECT) in Rotterdam, the Netherlands, is one of the most automated container terminal in the world.

However, most of the automatic cargo handling systems existing nowadays in ports and terminals have a common drawback: they use predefined fixed travel paths based on different guidance systems (such as rails, markers, wires or magnetic paths), which do not exploit the whole port open area and make vehicles to suffer from problems as collisions or deadlocks [6].

The presented work, which is inside the European Project *Cargo ANTs*, aims to take a step further in the safe freight transportation by creating free-travelling smart vehicles - such as Automated Guided Vehicles (AGVs) and Automated Trucks (ATs) - that can co-operate in highly dynamic shared workspaces as cargo port terminals. Specifically, this paper presents a real time, low cost and simple - yet robust - Detector and Tracking of Moving Objects (DATMO) algorithm based on single layer laser-scanner data. Although the amount of information of a single layer scanner is low, our previous experience in using it for localization, navigation [12] and tracking people in urban sites allows us to explore its use in port terminals. In this way, one of the motivations of this work was to extend the limits of actual 2D rangefinders, squeezing to the maximum the little information obtained, but still providing a robust and real time performance in challenging environments.

DATMO algorithms have presence in research of autonomous vehicles and urban robotics since long time. Typical algorithms takes the general approach (extended from Computer Vision algorithms) of dividing between object detection in static frames (fixed time) and tracking the detected objects along the time [9], [8].

Differently, the presented moving objects detector not only uses single frames, but generates an inter-frame object association to filter static objects, so that reducing the tracking load. This technique is of special interest in places as the port container terminals where there is not a predefined environment structure. Container terminals have no walls, buildings or signs, and containers are stacked in different ways in specific areas and these stacks can change in the same day. Moreover, the containers are transported by straddle carriers, AGVs or trucks and they can be found in any place in the port terminal which make even more complicate to differentiate what is moving and what is static.

Existing tracker algorithms perform associations between objects on different frames by means of closeness and propagate the tracked object positions. In [7], this propagation is done by using a linear and curvilinear constant trajectory, in addition with an unscented Kalman filter. Other approaches use hypotheses to choose the best movement trajectory of the object such as [4], which also uses an extended Kalman filter to estimate the object motion using a different model for pedestrians and vehicles.

In contrast, the presented tracking approach uses a multi-hypothesis algorithm for objects association. And a Kalman filter with linear constant velocity propagation is also used for simplicity to predict positions, as it has been checked that performs as well as curvilinear trajectory models due to the closeness of the detections in time, and reduce considerably the algorithm complexity. Furthermore, the presented tracking system improves the estimation of object motions by analysing them through a window of the associated previous tracks.

The rest of this paper is organized as follows. Section 2 details the components of the presented DATMO system. Section 3 shows the dataset obtained from a real port environment, as well as the experiments that support the system performance. Finally, conclusions are presented in Section 4.

2 Detection and Tracking Moving Objects in Ports

Common DATMO systems divide the process in two separate phases: firstly an object detector extracts objects from static frames of data (frame by frame), and secondly, a tracker algorithm takes the detected object and assigns a coherent label thorough the time to each object, creating the moving object targets.

The contributions of the DATMO approach presented in this paper are:

- it uses only a single-layer rangefinders sensors to detect and track objects, extracting valuable information from small amount of data. It also speeds up the tracking phase by pre-selecting moving obstacles in the detection phase.
- it is robust to highly dynamic shared workspaces as cargo port terminals, where the location of the containers can vary greatly, obtaining different port configurations: without containers, medium occupied with containers or fully occupied.
- the method is able to detect and track moving objects (straddle carriers without and with containers, automated trucks, AGVs empty or carrying containers, cars and people) that can have the same shape of the static objects (containers).
- it manages to keep targets of objects even when crossing paths, get occluded or are temporally not detected, by means of the multi-hypothesis tracking and its probabilities.
- it filters real static objects from moving ones by comparing its velocities with the vehicle velocity and by taking into account the constancy of the moving detected objects. Also groups separated detections from big moving objects in a singular one obtaining a better estimation of the real shape.
- it has been created and tested for free-travelling autonomous vehicles in port terminal, which are known to be well structured but also highly dynamic environments.

The next sections will explain in more detail both the moving objects detector and the tracking steps of the developed DATMO system to be used for AGVs in the port terminals.

2.1 Moving Objects Detector Description

The different modules of the moving object detector explained next, are shown in the general schema of Figure 1.

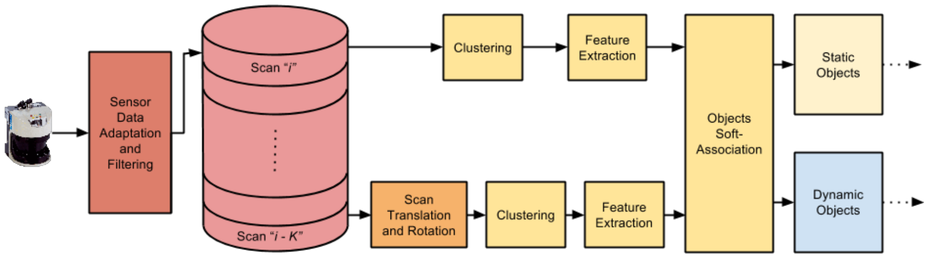


Fig. 1: General schema of the modules in the moving object detector.

Input Data Adaptation & Filtering. Every time i, a 2D laser scan is obtained, and points are filtered eliminating outlier samples (such as those coming from echoes of the laser-scan) as well as points beyond the interaction limit of the vehicle.

Scans Buffering. In contrast to other DATMO systems, this object detector is able to detect moving objects by comparing and matching different frames of the obtained data. These frames are hold in a buffer which size will depend on the applications of the detector. For the tested cargo terminal environment where vehicles speed is up to $6m/s$ (relatively slow) and the sensor provides data at $10Hz$, the buffer has been set to 10 frames so that, the actual objects will be compared with the ones detected 10 frames before (1 second).

Scan Translation and Rotation. Because the vehicle is moving, when the detector compares scan i with scan $i-k$, this last must be translated and rotated to be in the same actual reference frame. For doing this operation the detector use the information from the vehicle odometry. Note that odometry suffers from wheels drifting and other errors, but as in this case only the differential odometry between both scans is used, errors will not affect the algorithm performance.

Data Clustering. Groups of laser hits (clusters) are created by neighbours proximity. In single layer laser sensors, this can be done easily taking advantage of the provided ordered data, so that it is only needed to check the distance to the next point. However, when the detector use another sensor that provides unordered data, a k-nearest neighbours search must be performed in the euclidean space, keeping the simplicity and robustness of the algorithm.

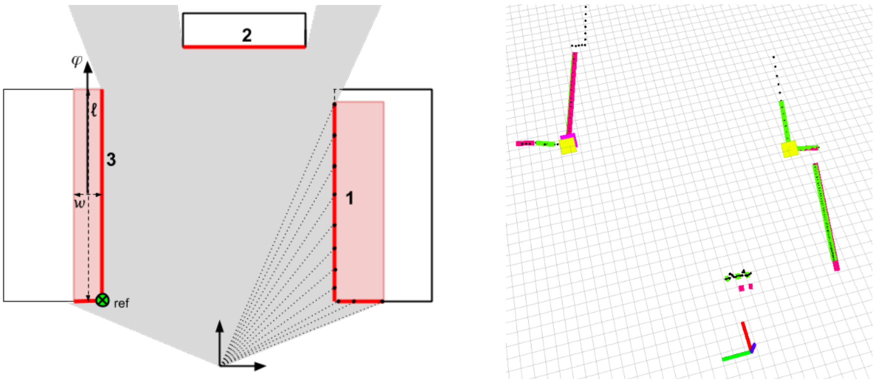
After filtering clusters by a minimum number of points and size, the resulting ones are considered as *objects*. The general object model created is a rectangular box as shown red in Figure 2a, represented by the following state vector:

$$obj_n(frame_i) = [ref_{act}(x,y), w, l, \varphi, np]^T$$

which contains respectively the reference point of the object, its width, length and orientation with respect the longitudinal axis, as well as the number of laser points contained.

Geometric Primitives Extraction. One of the problems of having only a single slice of data from the environment, is the absence of reference points in the partially observed objects.

Objects in port environments, as containers, buildings or vehicles are very internally structured and in most of the cases have well defined straight lines and clear corners. Therefore, a fast way to find common characteristics is to extract its inherent geometric primitives (such as lines or corners) as shown in Figure 2b. At this step, the model of the found objects will be completed with the lists of its detected lines and corners.



(a) Laser detection of objects.

(b) Lines and corners extracted.

Fig. 2: Examples on laser detections of objects and the geometric primitives extracted from real data after the odometry propagation.

Lines are found by means of linear regression over the groups of consecutive points. Then a validation procedure is performed to check if the error is kept low. In a second step, consecutive lines are fused attending to the angle between them. The identified lines are parametrized in a state vector by its initial and final points, its linear regression error, and its director vector as follows:

$$line_j(obj_n) = [init_p(x,y), end_p(x,y), error, vector]$$

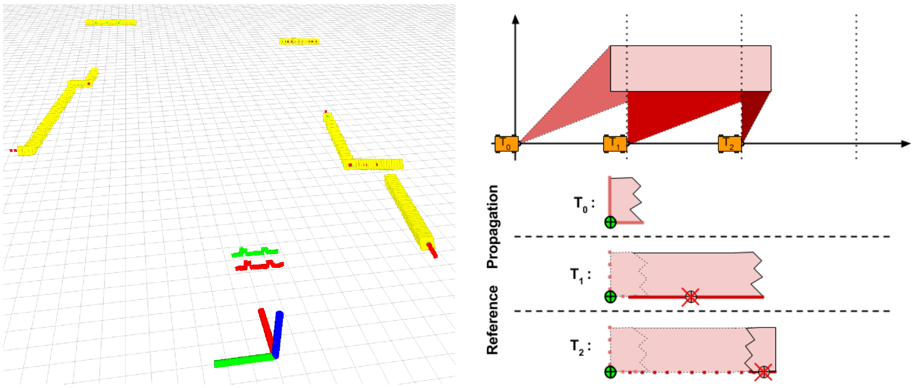
The corners are created when two lines cross at an angle larger than $\pi/6$, and are defined in a state vector by its position, orientation and aperture:

$$corner_j(obj_n) = [point(x, y), orientation, aperture]$$

Soft Object Association and Logical Reference Propagation. In this step, objects from the scan i and the scan $i-k$ under a common reference frame are compared to get whether or not an object has moved in time.

Each full object state vector (including lines and corners) of scan i is compared with every full object state vector of scan $i-k$ and its distances are measured. If the distance between an object in the actual scan with another in scan $i-k$ is less than an association threshold, it means that the object has not moved in time so that is tagged as static. Figure 3a shows an example of objects associated (in yellow) and a dynamic one (in red at the older scan and green the new, both in the common reference).

The soft association of objects allows also to propagate logical reference points for partially seen objects with no reference point, as well as information about size of the objects. The associations made by means of corners are considered very reliable and the corner position will be kept as the actual reference point of the object, even though it could not be seen in future frames. Figure 3b shows an example of how the reference of a corner is propagated over partial observations of an object (in T_1 and T_2).



(a) Associated static objects.

(b) Reference Propagation.

Fig. 3: Left, associated objects (yellow) tagged as static, and a dynamic detected object (in red the old detection and in green the current). Right, example of the actual reference propagation of a corner over associated objects with partial visibility.

2.2 Tracking and Filtering Moving Objects

The developed tracking algorithm for moving objects in port environments is based in the tracking approach developed by Reid [10], although the most probable hypothesis has been selected. Also, the tracker uses a similar approach for confirmation and elimination of the targets as the developed by [1], but in this case it has been modified to take into account the specificity of the port terminals. Moreover, it has been used the object tracking prediction of [5] and the local coordinates system described in [3].

Although this paper is focused in port environments, the tracker here explained has been also tested for people following in urban environments. A general schema of the tracker processes is shown in Figure 4. The presented implementation has several new functions and characteristics which will be explained in the next paragraphs.

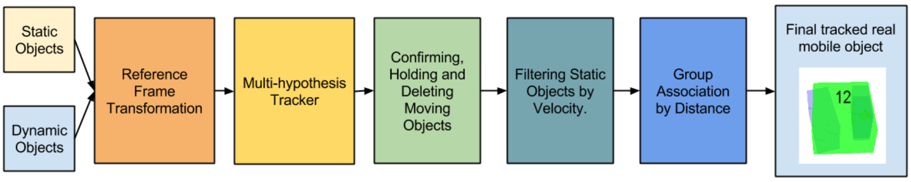

Static Objects
Dynamic Objects
Reference Frame Transformation
Multi-hypothesis Tracker
Confirming, Holding and Deleting Moving Objects
Filtering Static Objects by Velocity.
Group Association by Distance
Final tracked real mobile object
12

Fig. 4: Modules included in the Tracker system.

Confirming, holding and deleting moving objects. The proposed tracker modifies the probability formula described by Reid in [10], as it can be seen in the equation 1. The first modification is to deal with the false positive detections (not real moving objects) and the second one to take into account the false negatives (non detected moving objects). These improvements allow to have a better control of the confirmation, hold and deletion of the moving targets.

The general formula of the probabilities taking into account the new two terms is (1).

$$\begin{aligned} P_i^k = \eta" \cdot P_{\mathrm{det}}^{N_{\mathrm{det}}} \cdot (1 - P_{\mathrm{det}})^{N_{\mathrm{TGT}} - N_{\mathrm{det}}} \cdot \beta_{\mathrm{fal}}^{N_{\mathrm{fal}}} \cdot \beta_{\mathrm{new}}^{N_{\mathrm{new}}} \cdot \\ \cdot \mathcal{N}_{detector}(Z_m - H\bar{x}, B) \cdot {P_i}^{k-1} \cdot f_{\mathrm{c}}(t) \cdot f_{\mathrm{d}}(t) \end{aligned} \quad (1)$$

Where, ${P_i}^k$ is the probability of the actual hypothesis; $\eta"$ is the normalization factor; N_{det} is the number of detections; N_{TGT} is the number of existent targets; N_{fal} is the number of false positives; N_{new} is the number of new targets; P_{det} is the probability value of the detection for the detector; $(1-P_{\mathrm{det}})$ is the probability of no detection for the detector; β_{fal} is the Poisson distribution that corresponds to the false alarm; β_{new} is the Poisson distribution that corresponds to the new target; $\mathcal{N}_{detector}(Z_m - H\bar{x}, B)$ is the Gaussian probability distribution of the detections for the detector; ${P_i}^{k-1}$ is the probability of the previous hypothesis. All these terms are explained in detail in [10], where the reader is referred for a deeply explanation. On top of it, the presented tracker introduce two new functions. The first one, $f_{\mathrm{c}}(t)$, is a *confirmation function* created to confirm the tracks as mobile objects, and the second, $f_{\mathrm{d}}(t)$ is a *deletion function* which aim is to slowly erase tracks that have no detection associated. The tracker behaviour for confirming or erasing tracks can be balanced by modifying the parameters of both functions, so that the algorithm can be generalized and used in different environments.

In order to reduce the number of false positives (not real dynamic objects), the *confirmation function* (2) allows to control whether a target is confirmed or not by growing the probability in a slower way than other tracker approaches.

$$f_{\mathrm{c}}(t) = \begin{cases} \beta_{\mathrm{c}} & if\ l = 1 \text{ and target no confirmed} \\ \beta_{\mathrm{c}} + (\triangle\beta_{\mathrm{c}}) * (l-1) & if\ l > 1 \text{ and target no confirmed} \\ 1 & if \qquad \text{target confirmed} \end{cases} \tag{2}$$

The *confirmation function* (2) only applies when the target is still not confirmed and allows a slow rising of the target probability until it is confirmed. In it, the parameter $\beta_{\mathrm{c}} = 0.02$ is the *confirmation constant* and is obtained experimentally for providing continuity to the target probability, which have to continue from the new target probability value of 0.5. The parameter $\triangle\beta_{\mathrm{c}} = 0.01$ (can be between $[0, 1]$) is the *confirmation increment* that grows up each time the target has been associated to one detection (counted by l, being $l = 1$ for the first association after the new target generation). A low value of $\triangle\beta_{\mathrm{c}}$ implies a slow probability growth.

In the case of false negatives (no detections of moving objects), the presented tracker holds the existent moving targets during short time by means of the new *deletion function* (3). The *deletion function* decreases slowly the probability so its targets take longer time to be deleted and therefore the IDs are kept longer. Figure 5b shows a real experiment where the tracker is able to hold a track when no detection exists. If the detection reappears and is associated to the target, it keeps its ID.

$$f_{\mathrm{d}}(t) = \begin{cases} (\beta_{\mathrm{d}})^{\triangle t}\ if & \text{target no associated} \\ 1 \qquad if & \text{target associated} \end{cases} \tag{3}$$

In the *deletion function* (3), $\triangle t$ is the increment of time in seconds without detection associated to the target and $\beta_{\mathrm{d}} = 0.99$ is the parameter that allows the slow decreasing of the target probability. The behaviour of this function is like a negative exponential that decrease very slowly.

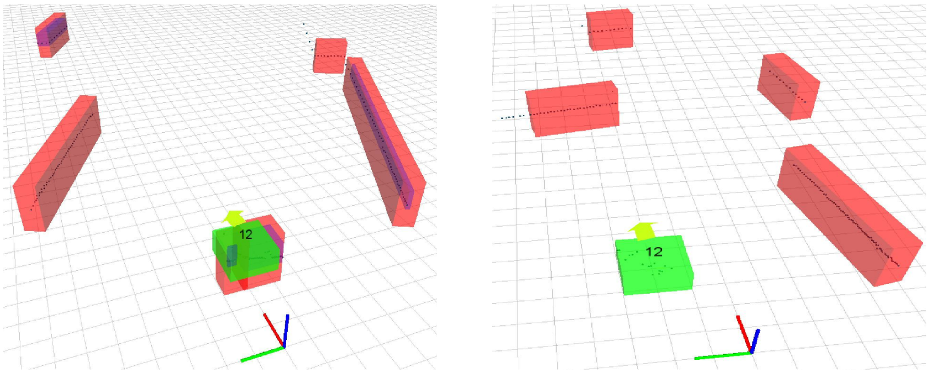
Filtering Static Objects by Velocity. Apart from the information about the movement of the objects provided by the detector, the tracker algorithm filters static objects that have global velocity 0. This is equivalent to filtering objects which linear velocity, as seen locally by the vehicle, is equal to $\overrightarrow{V}_o^{\,v}$, obtained by means of translation and rotation: $\overrightarrow{V}_o^{\,v} = -(\overrightarrow{V}_R^{\,w} + \overrightarrow{W}_R^{\,w} \times \overrightarrow{r})$; where $\overrightarrow{V}_R^{\,w}$ is the linear velocity of the vehicle in the world coordinates, $\overrightarrow{W}_R^{\,w}$ is the angular velocity of the vehicle in the world coordinates and $\overrightarrow{r}$ is the distance between the vehicle and the tracked object.

Group Association by Distance. The targets of the same type are included in the same group using the following grouping distance (4), this equation is the same that uses [10], but changing the detection by a target.

$$(H\bar{x_1} - H\bar{x_2})^T(H\bar{P_1}H^T + H\bar{P_2}H^T)^{-1}(H\bar{x_1} - H\bar{x_2}) \leq \eta^2 \tag{4}$$

Where, $\bar{x_1}$ and $\bar{x_2}$ are the target states and P_1 and P_2 are its corresponding covariances; H is a measurement matrix defined in the same way as in [10]. Here, η^2 have the same meaning that in [10] but with biggest value, because is associating different targets of the same moving object.

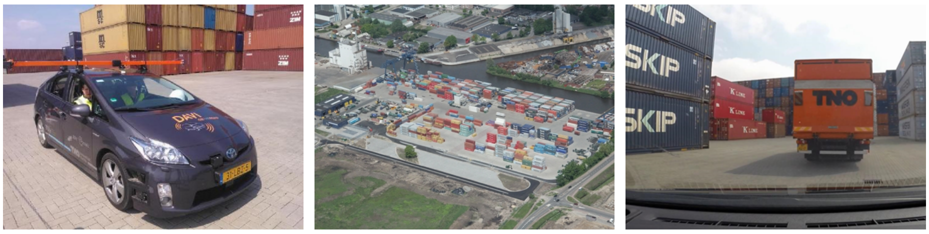
When the tracker groups targets of the same moving object that are detected in a separate way, it takes into account the shape of all of the grouped targets to get the real shape of the moving object. In this case, a new ID for the final group is created, as it can be seen in Figure 5a.



(a) Tracker filtering static objects and grouping dynamic objects.

(b) Tracker holding the real moving object in false negative situations.

Fig. 5: Tracking grouping and holding track examples over real data from a port terminal. In green the real mobile target with its identifier (12). Also, the red objects are the detections, the blue ones are the detection considered like a moving objects, and the reference frame represents the vehicle.



(a) TNO test vehicle setup. (b) Hengelo CTT terminal. (c) Vehicle following case.

Fig. 6: TNO test vehicle containing all the sensors used, CTT environment and image of the front view with the corresponding output of the algorithm.

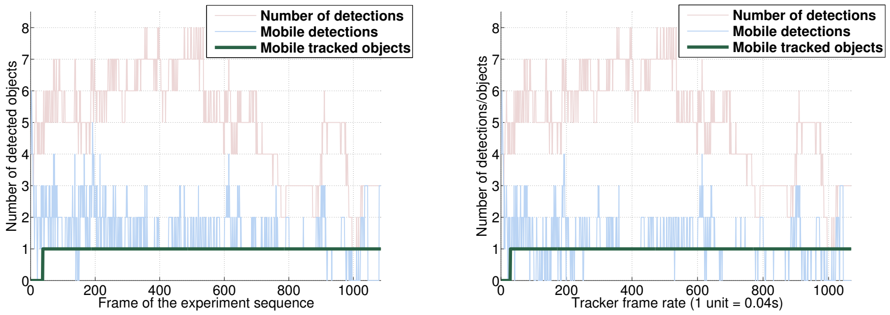
3 Experiments

This work has been perfomed under the Cargo ANTs project, and a dataset has been created for testing the developed algorithms. Real data from the CTT port terminal in Hengelo, the Netherlands (Fig. 6b), has been captured using the TNO test vehicle that included specific sensors for this project (Fig. 6a). The dataset includes relevant data for *vehicle following* inside the port. For this dataset, a truck has been chosen as target (Fig 6c), because Cargo ANTs is about a mixed situations of AGVs and automated driving trucks. The sensor setup of the TNO test vehicle is shown below:

- A TNO test vehicle from where the host tracking signals were logged (e.g. longitudinal velocity and acceleration).
- 3 Automotive radars (on top of the vehicle) which outputs are both object clusters and trackers.
- 6 Ibeo LUX laser scanners, mounted on the TNO test vehicle, capturing a 360 deg field of view as a pointcloud data.
- Accurate GPS measurement system of OxTS.
- Video camera which captures the front view of the TNO test vehicle.

For the purpose of the research described in this article, only the odometry measurements and a single layer laser scanner from the front view of the TNO test vehicle. Our results are extracted comparing directly to the ground-truth of the provided dataset, which contains 1065 frames of the vehicle following a truck. Two different DATMO system configurations, have been tested in the same dataset, a permissive case, and a restrictive one, which results can be seen in Fig. 7.

The first DATMO approach employs a more permissive detector with lower association threshold that introduce more false positives (static object considered as moving ones), as can be seen in blue in Figure 7a. Over the whole number of 4266 detections, 1539 are considered as dynamic which translates on a 10.95% of false positives and 6.91% of false negatives on moving objects over all the detections. This version uses a tracker configuration focused on filtering those



(a) Permissive detector and Tracker object filtering configuration.

(b) Restrictive detector and Tracker holding configuration.

Fig. 7: Results for two approaches of the DATMO system on a vehicle following scenario. Global number of detections are presented in red, dynamic ones in blue and final dynamic objects tracked in green. It can be seen how the restrictive case is more stringent when tagging objects as dynamic,and how the tracker is able to keep track of the vehicle even when no detection is provided.

false positives to get the real dynamic detections, which output accomplished the task of following the moving vehicle as shown with the green line.

The second version has a bigger association threshold on the detector phase, and therefore is more restrictive on detecting moving vehicles. It generates less false positives (reduced to 5.09% of all the detections), but introduces more false negatives (moving objects not detected - 13.09%), which are compensated by a tracker focused on keeping track of the true positives. The results of this version can be seen on Figure 7b.

In both cases there exist a configurable initialization period for initializing targets of the objects, imposing a trade-off between performance and initialization time - the longer initialization time, the most real static objects will be filtered and better performance will be obtained. In the permissive approach, an initialization time of 1.5 seconds shows very good results being able to filter correctly all the static objects and keeping track of the moving vehicle. For the restrictive case, as the tracker trust more in the detections and does not have to filter them so much, an initialization time of less than 1 second is enough.

4 Conclusions

This work describes a detection and tracking system for moving objects for AGVs in port container terminals using single layer laser scanner.

The detection algorithm is able to recognize the objects of the port container scenario including its real size. It has been defined two types of detector configurations, one more restrictive than the other, to study the performance of reduction of false positive and false negative.

The developed tracker allows to track only the correct dynamic objects, by eliminating all the false positives. It is also able to keep the target when there are false negatives (no dynamic objects detections). The tracker is able to group the different detections of objects of the same type, thus creating a target of the whole real object with its real shape. Finally, there have been verified two types of tracker configurations, corresponding with the two detector configurations.

The system has been evaluated with real data obtained in the CTT port terminal in Hengelo, the Netherlands. The results shows that the DATMO system accomplish the vehicle following task although little information provided by one single layer laser scanner has been used.

Acknowledgments. This work has been partially funded by the EU project CargoANTs FP7-SST-2013-605598 and the Spanish CICYT project DPI2013-42458-P.

References

1. Arras, K.O., Grzonka, S., Luber, M., Burgard, W.: Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. In: IEEE International Conference on Robotics and Automation (2008)
2. Carlo, H.J., Vis, I.F., Roodbergen, K.J.: Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. European Journal of Operational Research 236, 1–13 (2014)
3. Corominas-Murtra, A., P.J.P.S.: Multi-target & multi-detector people tracker for mobile robots. In: IEEE European Conference on Mobile Robotics (2015)
4. Ess, A., Schindler, K., Leibe, B., Van Gool, L.: Object Detection and Tracking for Autonomous Navigation in Dynamic Environments. The International Journal of Robotics Research 29(14), 1707–1725 (2010)
5. Ferrer, G., Sanfeliu, A.: Bayesian Human Motion Intentionality Prediction in urban environments. Pattern Recognition Letters 44, 134–140 (2014)
6. Kim, K.H., Jeon, S.M., Ryu, K.R.: Deadlock prevention for automated guided vehicles in automated container terminals. In: Container Terminals and Cargo Systems: Design, Operations Management, and Logistics Control Issues (2007)
7. Kim, Y.S., Hong, K.S.: A tracking algorithm for autonomous navigation of agvs in an automated container terminal. Journal of mechanical science and technology 19(1), 72–86 (2005)
8. Mendes, a., Bento, L., Nunes, U.: Multi-target detection and tracking with a laser-scanner. IEEE Intelligent Vehicles Symposium pp. 796–801 (2004)
9. Mertz, C., Navarro-serment, L.E., Maclachlan, R., Rybski, P., Steinfeld, A., Urmson, C., Vandapel, N., Hebert, M., Thorpe, C., Duggins, D., Gowdy, J.: Moving Object Detection with Laser Scanners. Journal of Field Robotics pp. 1–27 (2012)
10. Reid, D.: An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control 24(6), 843–854 (1979)
11. Roodbergen, K.J., Vis, I.F.: A survey of literature on automated storage and retrieval systems (2009)
12. Trulls, E., Corominas Murtra, A., Pérez-Ibarz, J., Ferrer, G., Vasquez, D., Mirats-Tur, J.M., Sanfeliu, A.: Autonomous navigation for mobile service robots in urban pedestrian environments. Journal of Field Robotics 28(3), 329–354 (2011)
13. Vis, I.F.A.: Survey of research in the design and control of automated guided vehicle systems. European Journal of Operational Research 170(3), 677–709 (2006)