Proceedings of 7th Transport Research Arena TRA 2018, April 16-19, 2018, Vienna, Austria

Autonomous driving on freely accessible railway tracks

Burkhard Stadlmann a*, Thomas Penkner a, Stefan Grünberger a, Stefan Dumberger a, Johann Weichselbaum b, Thomas Kadiofsky b, Wolfgang Pointner b

[a]*University of Applied Sciences Upper Austria, Stelzhamerstraße 23, 4600 Wels, Austria*
[b]*AIT Austrian Institute of Technology, Donau-City-Straße 1, 1220 Vienna, Austria*

Abstract

Autonomous vehicles are a rapidly emerging technology that will radically transform the face of public and personal transportation in the near future. The project autoBAHN2020 aims to promote this trend on local railway tracks to make them more attractive for passengers again. Trains, by concept, come with major shortcomings such as lack of privacy and inflexibility due to long cycle times and continuous stops. This problem could partly be overcome by splitting them up into multiple smaller *trainlets* while significantly reducing the cycle time and the number of passengers per trainlet. However, a train driver for each small trainlet would not be economically feasible, therefore a fully autonomous operation is required. Since many railway tracks are freely accessible, a sensor system for reliable obstacle detection is required. This technical paper will present the current state of the project autoBAHN2020 and its actual topics of research.

Keywords: autonomous train; ATO; driverless train; railway; sensor fusion; obstacle detection; simulation

1. Introduction

In order to increase the attractiveness of traveling on local railway tracks, the cycle time as well as the number of passengers per train should be reduced significantly. Less passengers per train directly leads to less stops as well as a more fluid travelling experience, resulting in an overall optimized travel time. To ensure operation at full capacity again trains need to be split up into multiple smaller trains, further referred to as trainlets. However, a train driver for each small trainlet would not be economically feasible and therefore a fully autonomous operation is required. Many of the currently available automated/autonomous railway transport systems such as the Copenhagen Metro (Vuk, 2005) or the Nuremberg U-Bahn (Briginshaw, 2004) promote economic efficiency and environmental friendliness but are all designated to be operated on special guide-ways instead of freely accessible tracks. The resulting massive initial costs would not be affordable for small local railway tracks. Further projects like the *Cargomover* from Siemens (Mairhofer, 2004) or the *Innovativer Güterwagen* from Deutsche Bahn AG (Dorn, 1997) intended to operate on freely accessible tracks but were all denied an approval to become a final product. One reason probably was that these autonomous trains would have been operated in a mixed mode along with regular trains resulting in additional operational, technical and economic barriers. In the field of passenger transportation, the project *RailCab – Neue Bahntechnik Paderborn* (Henke et al., 2007) presented a new concept of automated railway traffic with single passenger cabins. However, research was focused on mechanical solutions and to our knowledge an object detection system was not installed. Furthermore, *RailCab* would utilize a passive switch combined with an active vehicle steering, which would require the replacement of all switches on existing tracks. To overcome these barriers the autoBAHN2020 project aims for an autonomous system on freely accessible local railway tracks. Typically, those are closed systems with only one connecting point to main lines. Therefore, mixed traffic operation is not required when migrating to a fully autonomous system as described in Gebauer et al. (2012). Adding an intelligent sensor system for obstacle detection to each trainlet eliminates the necessity of expanding existing infrastructure in order to avoid collisions. As a consequence, autoBAHN2020 trainlets can easily operate on existing tracks without major investment. Additionally, each trainlet is equipped with an on-board localization system and a radio communication device ensuring a stable connection with the central operator station for data exchange. This allows a permanent monitoring of all vehicles and provides remote control functionality in case of failure or emergency.

2. Navigation

In regular operation, a trainlet is instructed to move from its current location to a specific destination. In order to execute this task, its position within the network and the route to the given destination has to be known. While finding a suitable route as stated in a movement authority for the train is the responsibility of the control center, localization is done by each trainlet itself. In addition, the exact pose is also essential for obstacle detection (see chapter 3), since this component has to determine if an object is within the structure gauge and therefore poses a higher threat concerning possible collisions. Each sensor is processing objects in its own sensor coordinate system, while the railway track, i.e. given route, is given in world coordinates. Therefore, all objects have to be transformed using exact localization information before running further collision tests.

2.1. Route map

The whole entity of railway tracks the autonomous train might be travelling on is represented within a single route map. This is a graph-based data structure storing all track segments together with the correct transitions between them. The map of the test track and the according graph representation are illustrated in Fig. 1. For the obstacle detection and localization tasks, precise knowledge about the 3D geometry of the tracks is mandatory. Furthermore, different local attributes related to autonomous driving tasks can be stored, e.g. positions of stops, crossings and signals as well as where to activate the train horn. During online operation, the information stored in the track data structure is accessed by multiple different software modules. Hence, efficient implementations are necessary for satisfying the constraints of these real-time applications.

2.2. Localization

For the operation of an autonomous train on railway tracks with inclinations and track cants, an accurate localization has to be determined by fusing multiple sensor signals into an optimal estimate of the trainlet's pose

Fig. 1 (a) track map example; (b) close-up of train station; (c) graph topology of the route map

in six degrees of freedom. Current sensors in use are GNSS (global navigation satellite system), inertial sensors, odometer, visual odometry and a balise reader. As trains are bound to the railway tracks, the route map assists correcting GNSS measurement errors and significantly increases accuracy of the lateral position. However, longitudinal misalignments are still a problem and can only be corrected partly by mounting balises along the track. Furthermore, magnetometer based compasses have been proven to be rather unreliable when used in environments surrounded by metal parts or strong electromagnetic fields. The only source of absolute yaw angle is derived from the track map, but possible longitudinal misalignments will lead to small angular errors again.

As stated before, in order to check for collisions, all objects have to be transformed into the world coordinate system first. Even small deviations in the estimated pose, especially angular errors, result in a significant transformation error, so that objects might appear at wrong positions. In order to overcome this challenge, a visual track detection algorithm (see chapter 3.2) provides additional input for correcting imprecise train orientations. The information of all sensors is fused with an extended Kalman filter (EKF), implementing a system model with a fifteen dimensional state vector.

3. Obstacle detection

3.1. Sensors and practical limitations

Due to the requirement of operating on freely accessible tracks, a reliable obstacle detection system is essential in order to correctly estimate oncoming collisions and react accordingly. To meet the high safety standards

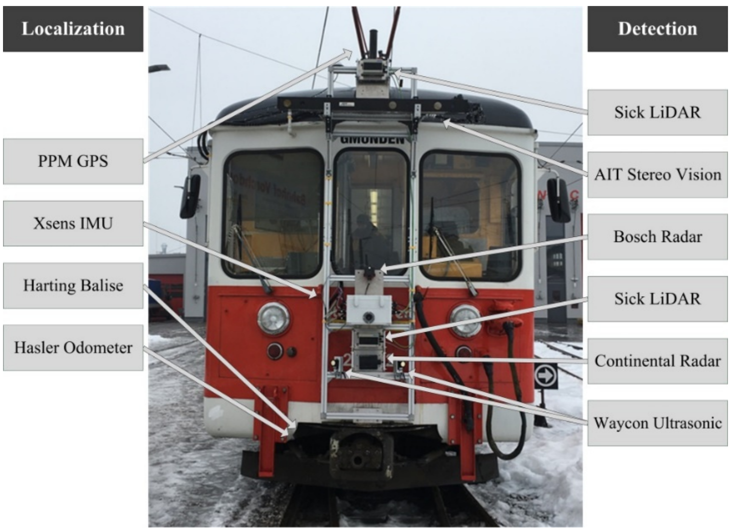

Fig. 2 demonstrator train used for field tests defined by the railway industry, a system of multiple sensors, each underlying different physical effects, is used. While laser scanners and visual sensors (Weichselbaum et al., 2013) perform best under common weather conditions, radar technology is more suitable when dealing with heavy rain or dense fog, but only in case of metal obstacles. A problem in our specific application has been found to be switches, as their surface areas facing the sensor might reflect a sufficient amount of radar signal to imitate a real obstacle blocking the path. Applying state of the art machine learning algorithms helps to partially suppress this effect, but does not solve it entirely. Despite its short range, ultrasonic sensing has been proven to reliably detect any kind of hindrance in the very near field and is therefore used at slow speed. It is essential for covering blind spots in close proximity to the train. Thermal infrared imaging is well suited for detecting living objects, even at night, and adds valuable information to the overall system regarding safety of pedestrians and animals. The following sensors are currently in use on the demonstrator vehicle (Fig. 2):

- 2x LiDAR
- Stereo Vision (visual)
- Stereo Vision (infrared)
- 2x Radar
- 2x Ultrasonic

Table 1 illustrates the experiences concerning single sensor performances gained from multiple field tests under different environmental conditions. It should be noted that the stated detection ranges might vary with different weather conditions and object types. The used LiDAR for example would detect a car in over 100 m distance, a pedestrian on the other hand only up to approximately 60 m. Extreme weather scenarios such as dense fog might further reduce its range to as low as 20 m, whereas a radar sensor would still detect cars up to 100 m.

Fig. 3 shows two challenging scenarios confirming the necessity of each single sensor technology. The first figure (a) shows a car attempting to cross the railway track ahead. Due to dense fog, the car is only visible up to

Table 1 sensor comparison regarding external influences

criteria rating: 1: weak / 4: very good	LiDAR	visual stereo camera	thermal infrared stereo camera	Radar (near range / long range)	Ultrasonic sensor (array)
horizontal angular resolution	4	4	3	1	2
vertical angular resolution	4	4	3	1	1
distance resolution	4	3	2	3	4
horizontal coverage / field-of-view	4	3	2	3/2	3
vertical coverage / field-of-view	2	3	3	2	2
coverage of object types & materials	3	3	2	1	3
support for tracks detection	2	4	3	1	1
support for object classification	2	4	2	1	1
tracking of relative speed of object (lateral)	3	3	2	1	1
tracking of relative speed of object (longitudinal)	3	3	2	4	1
robustness to environmental conditions: bright sun / night	3	2	3	4	4
robustness to environmental conditions: rain, snow, fog	2	2	2	4	3
estimate for useable range (meter)	~100	~100	~60	~40 / ~150	~5

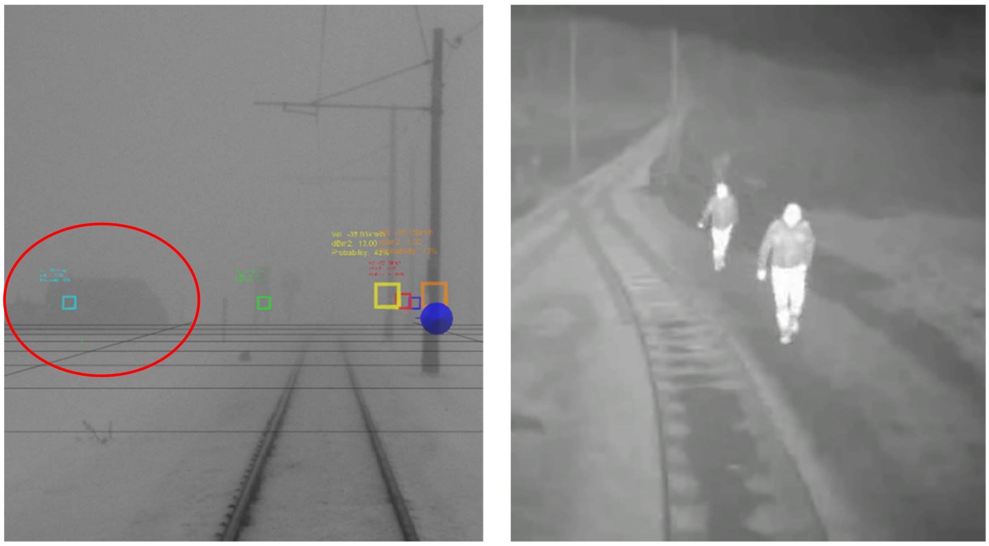

Fig. 3 (a) radar sensor with dense fog and crossing car; (b) IR camera and pedestrian

a distance of 50 m, but can be detected at a range of more than 100 m using radar technology. The second example (b) shows two pedestrians walking next to the track at night, clearly visible using thermal infrared imaging.

3.2. Visual track detection

The visual track detection is a key component for determining the exact position and orientation of the trainlet relative to the track in real time. Both visual features and 3D information from a proprietary stereo vision system, as first described in Zinner et al. (2008), are used to generate an accurate model of the track section that lies ahead of the vehicle. The applied method uses clothoid-based hypotheses generation and frame-to-frame tracking in order to achieve a stable and reliable estimate. The resulting visual track information given within the sensor coordinate system is used to improve the onboard localization by combining it with route information given in world coordinates. This fusion of local and global track information is essential for exact and robust far-range obstacle detection.

3.3. Object handling

In order to effectively combine all gained information and determine the most probable outcome, an intelligent sensor fusion is required. Furthermore, the object behaviour has to be modelled in order to predict future actions and act accordingly. State of the art machine learning algorithms for classification provide additional information as each underlying object class is bound to a specific behaviour. This data is provided for the subsequent reaction model described in chapter 4. The steps for object handling from initial detection to reaction is shown in Fig. 4.

All objects detected by the sensors are required to correspond to a predefined parameter structure, of which individual parameters can be masked out in case they cannot be estimated. A radar sensor for example might only measure an object's radial velocity and position, but does not have any information about its size or shape. It is the sensor fusion's responsibility to collect data from all sensors and fuse it in order to maximize the content of information and its reliability. The more often an object is detected by the sensors, the more certain the system will become and according actions can be initiated. An important property of the implemented fusion system is the ability to handle multiple objects simultaneously and to consider substantially different latencies inherent to different sensor technologies, as for example image processing algorithms tend to be rather time-consuming. In addition, an estimate for the dynamic behavior of objects is generated, which is essential for any sort of collision forecast. The underlying mathematical model also allows for predicting future object behavior, but uncertainty increases with time. The subsequent collision prediction estimates the probability of a collision with the trainlet in the near future, which will be the main input for the reaction model described in the following chapter.

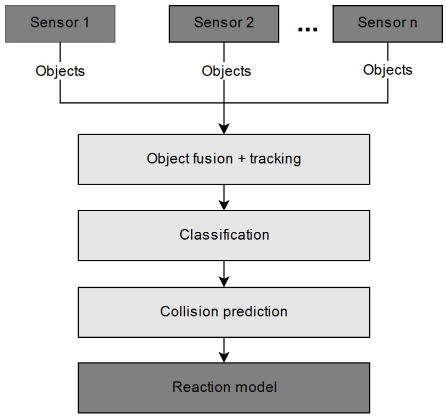

Fig. 4 object handling from detection to reaction

4. Driving strategy

A special driving strategy is essential for an economic, comfortable and, in particular safe transport of passengers. A train driving with an average speed of 5 km/h is just as inacceptable for passengers, as a train performing several emergency brakes, all caused by improper or unnecessary reactions in answer to detected obstacles. Therefore, several aspects need to be taken into consideration, such as knowledge of the trainlet's geographical route, its accurate position and speed, weather related environmental influences, as well as an elaborated trainlet reaction model regarding obstacles.

4.1. Risk analysis according to collision probability

In the risk analysis, we distinguish between static and dynamic objects. While the potential risk caused by static objects can be boiled down to the question if it is located in- or outside the trainlet's structure gauge, the situation for dynamic objects is much more sophisticated.

All detected dynamic obstacles are tagged with several parameters, which are necessary for a real-time risk analysis. The object fusion unit continuously provides a list of objects together with the time of detection, as well as the following set of parameters:

- Object class
- Object age
- Probability of collision
- Track meters to collision point
- Key points provided by a route atlas (such as level crossings, stops)

Each object is assigned one of five predefined classes: small object, pedestrian, pole, car/truck and big object. These differ in the object's size and mobility (static or dynamic). An object's age is important, as the longer it is tracked, the higher is the probability of a true detection. The calculation of the probability of a collision and the remaining track meters to the collision point is based on train speed, position and the object's movement. Finally, the track map provides information about dedicated key points along the track, which are also considered for the risk analysis. Such key points are level crossings or stops for instance, where it is more likely for dynamic obstacles (e.g. cars, pedestrians) to move into the structure gauge.

Based on these parameters, each object is assigned to one of four risk classes (low, medium, high, very high), which finally result in different reactions defined by the reaction model.

4.2. Reaction model

First, the list of objects is periodically sorted according to the risk classes from very high to low. Then, each object is fed to a dedicated decision tree. Initially, the movement of the trainlet is analysed, resulting in two further branches: a departing trainlet or a trainlet that is already in motion. In Fig. 5 the reaction model is depicted exemplarily for departures. In case static objects (see box 1) are located off the track and outside the

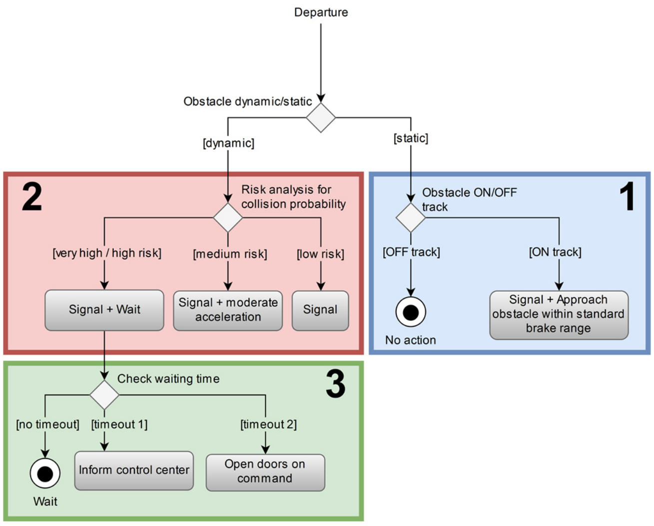

Fig. 5 decision tree branch for departing trainlets

structure gauge, no special action has to be considered and the trainlet can depart. Otherwise, the trainlet starts with triggering the horn signal. Furthermore, it approaches the obstacle slowly within standard braking distance and, in addition, an operator in the control center will be informed in case the problem persists.

Dynamic objects (see box 2) are evaluated according to their risk class. All subsequent branches have in common that a warning signal via horn is triggered. While for low and medium risk objects the trainlet can depart using limited acceleration curves and approach the obstacle at low speed, it has to remain stationary in case of high and very high risk objects. Moreover, there are predefined timeouts (see box 3) after which the control center will be informed and the doors of the trainlet will be opened on command.

5. Simulation

Early and frequent testing is a major component for the development of a reliable system. Thus, a simulation environment has been developed that supports virtual object detection and different simulation modes, such as generating custom scenarios or replaying recorded real world data. This allows testing algorithms and scenarios directly in the laboratory reducing costly and time-consuming field tests.

5.1. Simulation using recorded real world data

The first simulation mode is designed to rerun scenarios recorded during test drives. Thus, it is possible to play real world data repeatedly while still being able to pause or adjust parameters at any time. This type of simulation is especially beneficial for analyzing complex scenarios and furthermore optimizing algorithms. Recording raw sensor data provides the possibility for later adaptions of higher-level tasks and experimental implementations.

Fig. 6 (a) illustrates the architecture of the test system, in which the light grey blocks are part of the simulation environment and the dark grey ones are functionalities to be tested. In order to keep track of the enormous amount of data being processed in parallel, an essential part of the simulation is visualization. Using the wide range of built-in tools provided by the ROS (Robot Operating Systems) framework significantly reduces expenses for development. 3D visualization supports the user when interpreting spatial relations, while a camera overlay links abstract information to the real world scenery for quick visual validation. In Fig. 6 (b) the visualization of the 3D simulation including structure gauge, track map data and sensor data is illustrated.

Although testing with recorded data is useful for the development of some components, as it represents real world conditions, it is also very limited in some ways. First, the played back data from a past scenario includes the trainlet's reactions, which might change by adapting algorithms. Second, only scenarios recorded with a demonstrator vehicle can be tested. That's why collecting data of crashes or borderline situations in the real test environment is hardly possible. In addition, faulty behavior must be detected manually, since the simulation system is not suitable for inferring the correct reaction from the raw sensor data.

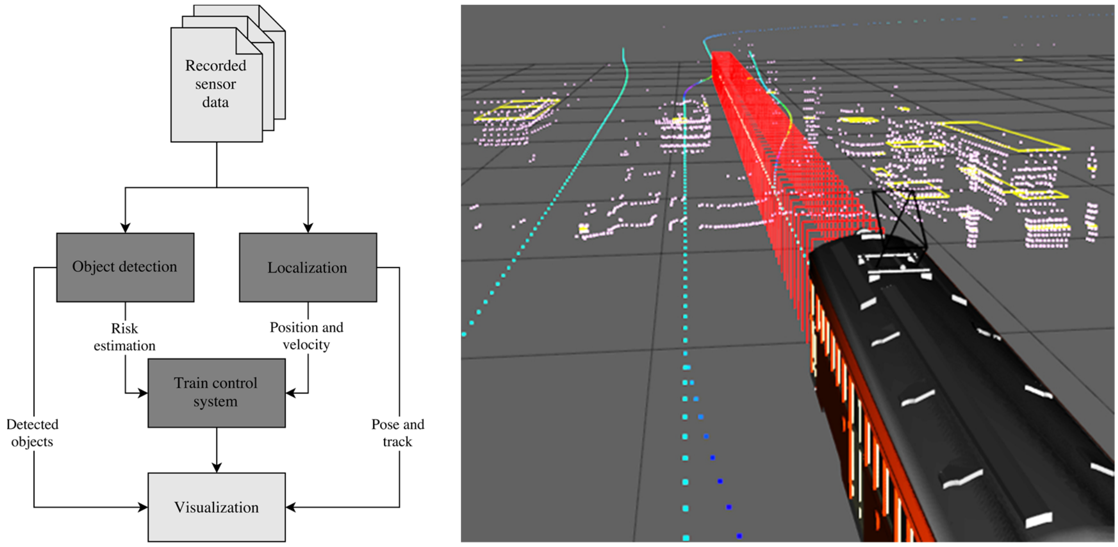

Fig. 6 (a) architecture using data from a test drive; (b) 3D data visualization

5.2. Simulation using virtual objects

Extending the simulation architecture from section 5.1 with the option to emulate sensors, allows the simulation of any possible scenario, even corner cases such as crashes. The extended architecture is illustrated in Fig. 7 with the same color coding as used in section 5.1. However, instead of simulating raw sensor data, it is sufficient to create virtual objects as perceived by each sensor and directly forward them to the system. The following features can be manipulated:

- Object size
- Trainlet/Object position
- Object lifetime
- Motion and dynamics
- Object detection by sensors
- Bias and noise
- Interaction with the trainlet

In order to simulate movements of a trainlet, a behavioral model based on the track map was implemented. Thus, the system directly specifies a desired velocity and the simulation will react accordingly. The movement, which

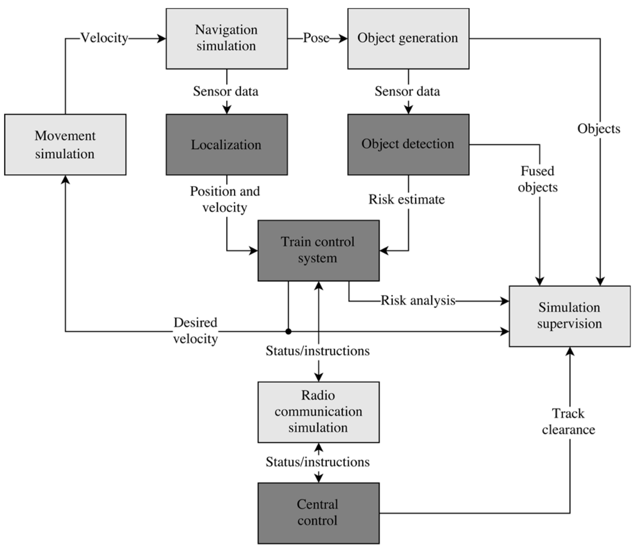

Fig. 7 architecture using virtual objects is bound to the track map, is the only limitation to the variety of possible test scenarios, but the map can be easily extended. In addition, the implemented communication channel with the central control station provides the possibility of a simultaneous simulation of multiple interacting trainlets.

The most important feature of this simulation architecture is the control and knowledge about every virtual object. Using this information, a supervising algorithm can compute a desired reaction to a certain situation, compare it with the trainlet's reaction and report faulty behavior. However, due to the high complexity of real world scenarios, only simplified versions can be represented by this approach.

6. Conclusion

The autoBAHN2020 project demonstrates the feasibility of autonomous trains on freely accessible routes and promises an increase in passenger numbers with lower CO_2 emissions thanks to the trend-setting trainlet concept. A state-of-the-art sensor concept guarantees obstacle detection under a wide variety of environmental conditions and even exceeds human perception in special scenarios. For future implementations, sensors with a higher resolution and range can further improve the final result to detect even smaller objects. In a further step, we will optimize the orientation estimation of the trainlet in order to detect fewer false positives at large distances.

In most cases, trainlets will require the complete sensor system on both sides, which is why the economic efficiency of these devices is a crucial point. However, various legislative adjustments are still necessary for final operation in regular traffic.

Acknowledgements

The project has been funded by the Austrian Climate and Energy Fund and the Austrian Research Promotion Agency (FFG).

7. References

Vuk, G., 2005. Transport impacts of the Copenhagen Metro, Journal of Transport Geography 13(3), pp. 223–233.

Briginshaw, D., 2004. First metro to convert to full automation. International railway journal and rapid transit review 44(9).

Mairhofer, F., 2004. CargoMover: An innovative mode of automated freight transport. Rail Engineering International 33(1).

Dorn, C., 1997. Innovative Güterwagen – Entwicklungsstand und Perspektiven. Der Eisenbahningenieur 48(8), pp. 7–13.

Henke, C., Rustemeier, C. Schneider, T. Böcker, J., Trächtler, A., 2007. RailCab – Ein Schienenverkehrssystem mit autonomen, Linear-motor getriebenen Einzelfahrzeugen. International ETG-Kongress.

Gebauer, O., Pree, W., Stadlmann, B., 2012. Autonomously driving trains on open tracks – concepts, system architecture and implementation aspects. it - Information Technology: Methoden und innovative Anwendungen der Informatik und Informationstechnik 54(6), pp. 266–279.

Weichselbaum, J., Zinner, C., Gebauer, O., Pree, W., 2013. Accurate 3D-vision-based obstacle detection for an autonomous train. Computers in Industry 64(9), pp. 1209–1220.

Zinner, C., Humenberger, M., Ambrosch, K., Kubinger, W., 2008. An Optimized Software-Based Implementation of a Census-Based Stereo Matching Algorithm. Advances in Visual Computing, Springer, pp. 216–227.