# An Obstacle Detection System for Automated Trains

Milan Rüder Nikolaus Möhler Faruque Ahmed
Fraunhofer Institute for Transportation and Infrastructure Systems
Driver Assistance and Operations Control
Zeunerstraße 38
Dresden, Germany
email [rueder|moehler|ahmed]@ivi.fraunhofer.de

### Abstract

*Safety is an important topic for any railway system. If trains are running on a track, which is not guarded of objects by fences or other means, special care for obstacles in front of the train needs to be taken. On a normal train, this is the task of the train driver. When implementing automated trains, a substitute is needed to ensure the same safety level as in normal operations. Therefore an obstacle detection system is required for safe operations. During the KOMPAS project we developed a system capable of monitoring the track and detecting objects in front of an automated train.*

## 1 Introduction

Railway systems are used all over the world. To increase the quality of service and to operate the trains on the need of the passengers, many attempts were made to automate the operations. Most of these attempts try to give help to a train driver. We call a system like this a driver assistant system, because it aids the driver while keeping him responsible of the situation. Assistant systems are widely used on normal tracks in many countries. A second type is a fully automated system. Implementations are installed all around the world. Most of the systems of this type operate on a dedicated track, which is guarded by fences or other means of protection against obstacles. Special equipment like doors is needed for each platform.

The German research project KOMPAS aims to automated railway systems, without the need of special, dedicated track. An automated system should observe the area in front of a train and detect objects on the track. We propose such a system which monitors the track in front of a train using a multi sensor setup. This setup consists of three video cameras and an infrared radar system. The sensors are connected to two PCs, which then run obstacle and track detection algorithms on the input. All detected objects are fed into a Kalman filter for data fusion and tracking. A decision algorithm determines if a tracked object is a dangerous obstacle. If an obstacle is detected, a message is issued to the automated vehicle control computer, which takes the appropriate steps to reduce the danger.

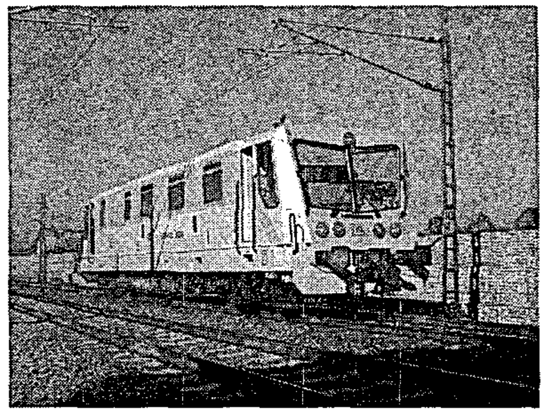

**Figure 1:** The test vehicle used during the development and test of our system.

The developed obstacle detection system was tested on a test vehicle of the German Railway as well as in normal operations on a suburban train. Figure 1 shows an image of the used vehicle. The tests showed detection results for small test objects of 0.4 $m^2$ in size up to a distance of 250 meters

The next section describes the used hardware for our system. In section 3 we cover the different obstacle detection algorithms, followed by the data fusion in section 4. Section 5 describes the experimental results of our testing.

## 2 Hardware

The obstacle detection system uses a multi sensor setup build from off the shelf components. No special sensors were used.

0-7803-7848-2/03/$17.00 ©2003 IEEE

Three progressive scan CCD video cameras are the main sensors. Two are used in a stereo setup with 0.35 $m$ baseline. They are fitted with lenses of 12 $mm$ focal length and are used for monitoring the near range up to a distance of approximately 50 meters. One camera with a 35 $mm$ lens is used for the far range. To cope with the changing illuminations of the environment, the brightness of the camera images is controlled by the computer. All cameras are fixed behind the windshield at a height of 3 $m$ above the ground.

Additionally a sensor originating from an adaptive highway cruise control system for cars is used. It is a multi beam infrared radar with a detection range of 150 meters and an opening angle of 8 degrees, mounted outside at the front at a height of 0.6 $m$. The mounting of the sensors can be seen in figure 2.

**Figure 2:** Mounting of the cameras and the infrared radar.

The obstacle detection system receives measurements of velocity and position data along the track from the vehicle control computer as well as from a differential GPS receiver.

Two PCs, each with two Pentium III processors at 1.0 GHz, are used for the system. The two computers are connected via an Ethernet. The connection to the vehicle control computer is an CAN bus.

An overview over the different components of the system and their connections is given in figure 3.

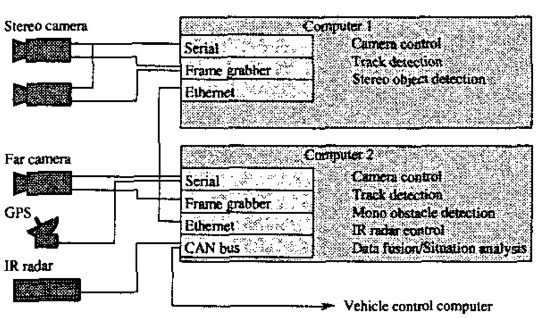


**Figure 3:** An overview over the obstacle detection system.

## 3 Image Sequence Processing

### 3.1 Track Detection

The first step in our system is to find the course of the track in front of the vehicle in 3D space. We use a clothoidal model [5] for the track that is given by the following equations:

$$c(l) = c_0 + lc_1 \quad .$$

$c(l)$ describes the curvature at the length $l$ of the clothoid. With our track detection we estimate the curvature $c_0$, the change of curvature $c_1$ as well as the yaw angle and the lateral position. To be able to model switches in the track the complete lookahead is divided into several clothoids, which have a smooth transition. The detection can be triggered to follow either way of a track switch.

Measurements for the estimation are found on search lines in the camera images. In figure 4 we show an image with the results of the track detection.

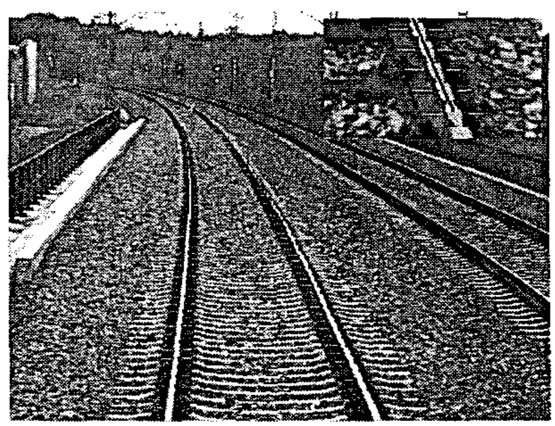

**Figure 4:** Result of the track detection. In the upper right corner is a magnification to better show the detected measure points.

### 3.2 Obstacle Detection

The obstacle detection is mainly done by image processing. Different algorithms are used simultaneously on the input images.

Not every algorithm detects all possible obstacles therefore we use the combination of these algorithms to stabilize the results:

1. Track gaps,
2. Properties of edge elements,
3. Gray value variance and correlation along the track,

4. Optical flow,
5. Statistics of textures and
6. Stereo by inverse perspective mapping.

All algorithms except the track detection run in parallel and calculate only on the part of the image which shows the track. The results are delivered to the data fusion module described in section 4.

**3.2.1 Track Gaps:** The first algorithm to find obstacles on the track is the gap detection. This step is performed in the track detection module. We check, if track measurements are found on all search lines. If there are some neighboring lines without points, we assume that an obstacle is hiding the rail or the rail is missing. Both are critical situations.

In figure 5 we show an image with a detected wood lying on the rail. This pole is hardly visible, because it resembles the wood in the track and gives a rarely different image, but it is clearly detected by the gap module.

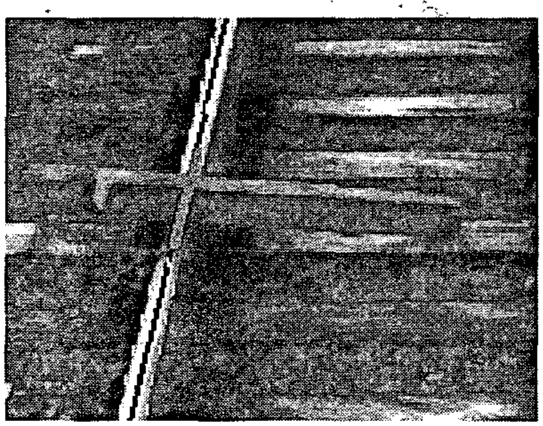

**Figure 5:** A wooden pole lying over one rail. The detection is shown as a bold orange bar.

**3.2.2 Edge Elements:** As a second clue for an object, we analyze edges in the image. The normal track has, apart from the two rails and the wooden supports, many small curved edges arising from the stones. We calculate an edge image with the Canny operator. We subtract all edges rail and support edges from this image. In the resulting image we search for long edges and build clusters around areas with many of these.

If the cluster face in 3D space is large enough, we assume an object. In figure 6 we show one example of this detection method.

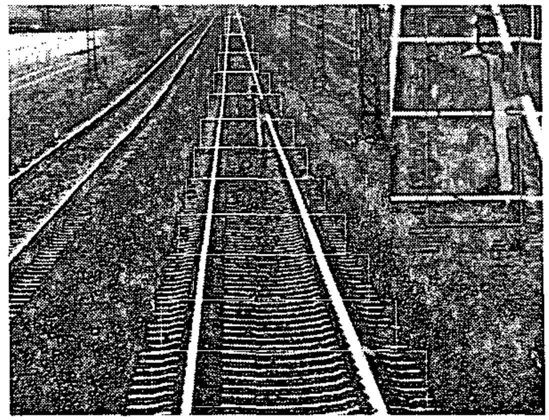

**Figure 6:** A person standing on the track. The detection is shown as a green box around the object. The red overlay shows the long edges, the white boxes the regions of interest.

**3.2.3 Variance and Correlation Analysis:** The third obstacle detection algorithm analyzes the gray value variance and correlation along the track. The idea behind this algorithm is that the track itself always has a similar visual appearance and gray level distribution. An object on the track will break this similarity in gray level distribution as well as in visual appearance. To be independent of the perspective distortion, a birds eye view of the scene is calculated. In this image we place consecutive rectangles on the track. In each of these rectangles the gray value variance is calculated. Then the variances of two consecutive areas are compared. If the difference is to high, an obstacle is detected.

As a second step we correlate each rectangle to it's neighbors using a mean value free, normalized cross correlation [1]. If the correlation is bad, we also have a hint to an object on the track. The image in figure 7 shows results of this algorithm as an overlay over the detected objects.

**3.2.4 Optical Flow:** A moving object in front of the train shows up in an optical flow field. This leads to our next algorithm. We calculate the optical flow field in our birds eye view image with a technique described by [3]. This simplifies the detection as the perspective distortion is removed and the vector field of the empty track should be uniform. From each calculated vector we subtract the motion vector of the train. After this step we have a vector field with nearly only zero vectors.

If an object is moving on the track, we have different vectors. A clustering around flow vectors of similar direction and length gives us a sign of an object. In figure 8 a result of the optical flow obstacle detection

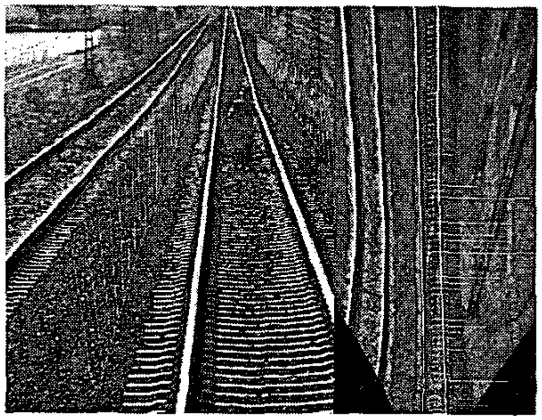

**Figure 7:** A person standing on the track. The detection is shown as a red boxes around the object. In light red the region of interest is shown. The right side shows the corresponding birds eye view with one line of rectangles in blue. The cyan vectors denote the probability of an object in the box, the longer the higher.

is displayed.

**3.2.5 Statistics of Textures:** As stated before, the track with it's stones has a specific gray value variance, which is broken by objects. A second algorithm exploring this property calculates the gray value variance in regions of interest placed on the track. Then the variance is calculated in small windows and compared with the expected value. This comparison leads to a binary image showing all pixels where the texture statistics changed.

Building a cluster around these pixels gives us an obstacle detection, if the cluster face is large enough. The images in figure 9 display the results of this calculations.

***3.2.6 Stereo Detection:*** *As a the last variant* of detection algorithms we use a stereo obstacle detector. We deploy the stereo effect by inverse perspective mapping of the camera images to the floor plane of the track. Since all cameras are calibrated to the same coordinate system, a difference of the mapped images results in a zero image if the plane assumption is valid.

In regions of the image where objects are rising above the ground, we get a difference in our mapped images. We build clusters around these areas, check the cluster size and have our object hypothesis. Figure 10 illustrates this.

**Figure 8:** A person moving across the track. The detection shows red box around the object. On the right side a birds eye view with overlayed optical flow vectors is displayed.

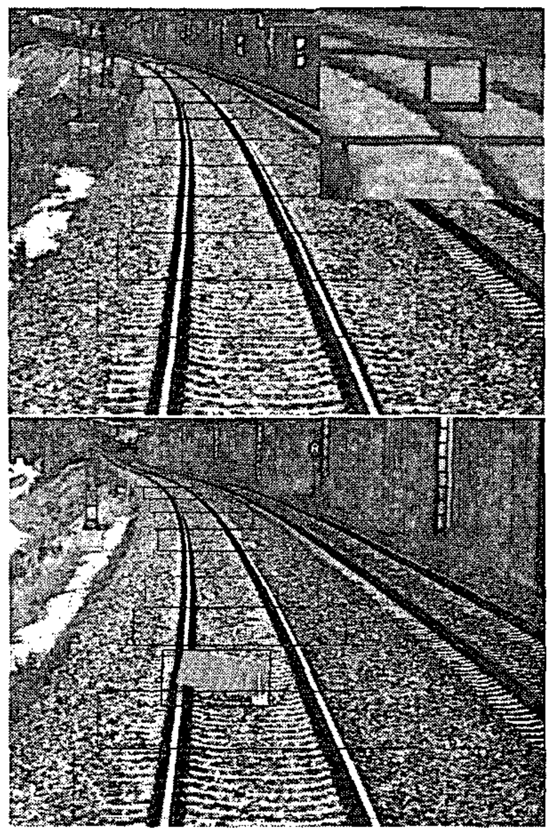

**Figure 9:** A small test target of 0.4 $m^2$ is placed on the track. The blue boxes are the regions of interest, the red box is the detected obstacle. In the upper image the target was around 250 meters away in the lower around 50 meters.

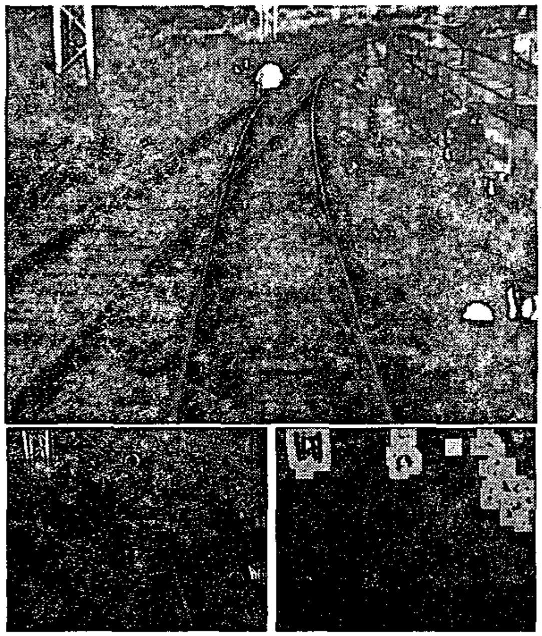

**Figure 10:** A person moving in front of the train. The upper image displays the left stereo image. Overlayed are the red obstacle boxes. The lower image shows on the left the differences during the inverse perspective mapping and on the right the resulting clusters.

## 4 Data Fusion

In this section we describe the data fusion. As noted above all detected objects are transmitted to the fusion module. This modules uses a Kalman filter [2] to fuse the objects to a single data base and to track the objects over the time. The filter estimates the position, the velocity and the acceleration of each object.

The update of a tracked object is done with the best matching object of each detection algorithm. Detected objects, which do not match a tracked object are used to initialize new tracked objects. If a tracked object is not supported by measurements it is deleted after some cycles.

All tracked objects are classified as belonging to the track, next to the track on a platform, an infrastructural object or aside the track. This classification is supported by a database, which records the infrastructure, such as platforms, bridges, poles for the electricity, etc.

If an object is of potential danger, it is reported to the vehicle control computer via a CAN bus. In figure 11 a visualization of the fusion is shown.

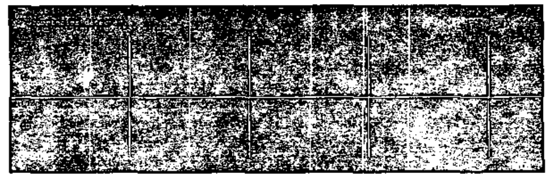

**Figure 11:** Visualization of the fusion with several objects in front of the train.

## 5 Results

Our obstacle detection system was tested on many different scenarios. It detected objects of the size of 0.4 $m^2$ up to a distance of 250 meters. A larger object would be detected from a greater distance. Persons wearing a warning waistcoat moving on the track were detected up to 300 meters. Only a minimum of false positives showed up.

The data fusion tracked the objects as the train approached the obstacle. Potentially dangerous objects were reported to the vehicle control computer which recorded all data for further analysis.

The detailed analysis of the data collected on the suburban train in regular service needs to be done. Only with a large database like this, a conclusion about the long term system behavior can be made.

## 6 Conclusion

We proposed an obstacle detection system to insure the operational safety of automated train systems. Up to now no such system is in use. We conducted long term test of our system on public tracks in regular service. The overall performance is promising.

Work that needs to be done includes a detailed error analysis to find out typical situations. Another field of work is the increasing of the detection range. For a typical train operation at speeds up to 100 km/h, the distance should be increased up to 600-700 meters. But even a human driver can not easily cover that range, especially on curved tracks or at railway stations. To cover the station we want to integrate our obstacle detection system with our platform supervision [4]. A combination of both can significantly increase the safety particularly at the highly dangerous platforms.

This work has been conducted during the KOMPAS project and was supported by the Germany Federal Ministry of Education and Research. Especially we want to thank the German Railways "Deutsch Bahn AG" and Alcatel SEL for their support in this project.

## References


[1] P. Aschwanden. *Experimenteller Vergleich von Korrelationskriterien in der Bildanalyse.* Hartung-Gorre, 1993.

[2] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association.* Mathematics in science and engineering. Academic Press, Orlando, Fl., 1988.

[3] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.

[4] W. Oertel, T. Dimter, and D. Szoska. A video-based approach for stationary platform supervision. In *Proc. of the IEEE Computer Conference on Intelligent Transportation Systems, Singapore, 3-6 September 2002*, pages 892–897, 2002.

[5] R. Risack, P. Klausmann, W. Krüger, and W. Enkelmann. Robust lane recognition embedded in a real-time driver assistance system. In *Proceedings of the IEEE Intelligent Vehicles Symposium '98, Stuttgart, Germany*, pages 35–40., October 1998.