Track detection for autonomous trains

Michael Gschwandtner[1], Wolfgang Pree[2], and Andreas Uhl[1]

[1] Multimedia Signal Processing and Security Lab (WaveLab)
Department of Computer Sciences
University of Salzburg, Austria
{mgschwan,uhl}@cosy.sbg.ac.at

[2] C. Doppler Laboratory Embedded Software Systems
University of Salzburg, Austria
wolfgang.pree@cs.uni-salzburg.at

Abstract. This paper presents a way to efficiently use lane detection techniques - known from driver assistance systems - to assist in obstacle detection for autonomous trains. On the one hand, there are several properties that can be exploited to improve conventional lane detection algorithms when used for railway applications. The heavily changing visual appearance of the tracks is compensated by very effective geometric constraints. On the other hand there are additional challenges that are less problematic in classical lane detection applications. This work is part of a sensor system for an autonmous train application that aims at creating an environmentally friendly public transportation system.

lane detection, autonomous vehicle, machine vision, train

1 Introduction

Autonomous transportation systems like ULTra[3] (Urban Light Transport) are trying to create an economically efficient and environmentally friendly transportation system. Most of the currently available (or under construction) Personal Rapid Transit (PRT) systems are designed to be operated on special guide-ways (i.e. railways) that are sometimes even elevated. Thus the implementation of such a system is bound to a huge amount of initial costs. The ULTra system states constructions costs per mile between £$5M$ and £$10M$. Another solution would be to operate autonomous cars on the streets along with the normal traffic. This however vastly increases the complexity of the scenarios that have to be handled by the autonomous cars. Furthermore, it is a substantial threat to the safety of passengers and outside traffic participants (especially children). The operation of autonomous vehicles on separate guide-ways seems to be a reasonable but expensive solution, because the path of the PRT vehicle does not conflict with the normal traffic and the track is predetermined which basically reduces the control commands to *stop* or *go*.

[3] http://www.ultraprt.com

A cost effective solution that interferes only marginally with normal traffic seems to be somewhere between the PRT and autonomous cars. We need a system that can drive along a fixed path but does not require the building costs for a whole dedicated guide-way/track. A possible solution to this is the use of existing railroad lines (smaller branch lines) that are then operated autonomously with smaller passenger trains (so called *train-lets* [3]).

Such a system obviously consists of many parts ranging from high-level trainlet control to low-level sensor data acquisition. In this paper, we are going to look at how lane detection known from autonomous cars can be used to aid obstacle detection and also to provide basic region of interest information for other sensors.

We present an algorithm that uses simple, yet efficient geometric constraints derived from the unique properties of *railroad tracks* to allow fast and robust track detection. The obtained position information can then be used for obstacle detection and sensor-fusion. In Section 2 we look at previous work that is related to lane detection. Section 3 compares the properties of streets with railroad tracks. The basic idea for lane detection adapted to autonomous trains is explained in Section 4. In Section 5 we present several examples of working and problematic scenarios. We conclude with an outlook how this system can be potentially improved in Section 6.

2 Previous Work on Lane/Track Detection

Lane detection for driver assistance systems is a topic that gained a lot of attention during the last ten years ([2,4,5,6,7,11,12]). Some approaches work on the acquired images directly which represent a perspective projection of the scene ([2,11,1]) and some perform a conversion of the scene into a top down view called Inverse Perspective Mapping ([4,5,7,9]). Most systems use a simple lane model to describe the lane which also applies to railroads. However only few systems are designed specifically for railroad detection ([8,10]). An obstacle detection system for trains is proposed in [10]. In [8] a vision based system for collision avoidance of rail track maintenance vehicles is proposed. It is based on detecting railroad tracks by applying techniques similar to lane detection for driver assistance systems. The spatial period of the sleepers and the distance between the rails are used to calibrate the camera parameters. A piecewise quadratic function is then fitted to candidate rail-pixels in the acquired (perspective) image and compared to the previous frame. However to the best of our knowledge no fully functional vision based obstacle detection system for railways exists to date. This work uses the well researched field of lane detection and tracking and extends it to the field of train applications.

3 Comparison

While the basic task of lane detection in street and railway scenarios is the same, there are several properties that require special attention and may help us to improve the robustness of automatic train systems. Table 1 lists some important differences, which are discussed subsequently.

Street	**Railway**
variable lane width	fixed lane width
variable lateral offset	zero lateral offset
varying type of lane markings	fixed “lane markings”
general lane appearance is relatively homogeneous	several different (inhomogeneous) “lanes”
lane markings are designed for optimal visibility	visibility is not guaranteed
lane markings have no volume and thus don’t cast shadows on the ground	tracks have a certain height and thus cast shadow on the ground
construction sites and obstacles can easily change the path of a car within a road	vehicle path through the world coordinate system is fixed
the speed of a car is generally adapted to weather and visibility conditions	automatic trains are operated at nearly constant speed independent of most weather conditions
the horizontal movement of a car-mounted camera is low due to the low height of the vehicle	swinging of the train cause substantial change of the camera position along the path

Table 1: Properties of street lanes and *railway tracks*

First of all, the lane width on railways is obviously fixed along the whole path. Otherwise the train would not be able to drive the complete track. The width of lane markings on streets, however, depends primarily on the type of road and is generally limited by a minimum width required by law, which in turn depends on the country the street belongs to. The lateral offset of a car relative to the center of the lane is variable, which is especially true for wider lanes. We will later see that a fixed lateral offset can be exploited to reduce the possible track candidates.

Road lane markings are designed in a way that should be optimally visible to a human observer (Figure 1b), and they are continuously improved. In contrast, the only purpose of railroad tracks is to guide the train. It is just a side effect if the tracks are easily visible to an observer. Although in many situations the tracks are very prominent, in general visibility is affected by changes in lighting and weather much stronger than road lane markings. An advantage of rails over lane markings is that they are constantly grinded each time a train rolls over them. This keeps the top of the rails from corroding. The appearance of the street itself is also very homogeneous (Figure 1b) compared to the track bed of railways (Figure 1a). The track bed, or in general the space between the rails, consists for example of gravel, asphalt, snow or even grass (Figure 1a). The last significant difference in the visual appearance is the volume of the tracks. Lane

(a) Inhomogeneous lanes between railroad lines (average case scenario)

(b) Good visibility of lane markers and very homogeneous roads

(c) Summer with vegetation

(d) Bad lighting because of shadow

(e) Summer no vegetation

(f) Good winter conditions

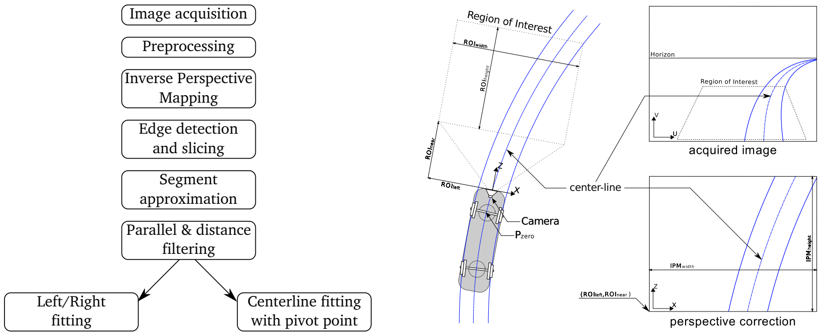
Fig. 1: Comparison between road lanes and *railway tracks*, and overview of different scenarios

markings are flat and basically have no volume or height. This means that they can not cast shadows on the ground and thus the detection of the lane marking itself is invariant to the position of the sun, if no other object casts a shadow on the street. Tracks however have a certain height, i.e. several centimeters, and thus cast shadows which create additional edges in the image and *weaken* the visual appearance of the real edge.

If we combine the fact that the lateral offset is fixed and that the position of the track can not be changed easily, it is clear that the train always moves on a fixed path with respect to the world coordinate system. This allows a much tighter integration of a-priori information like the prediction of the vehicle position at upcoming points in time. However this is only true for the whole vehicle, since trains have a strong trend to swing left/right especially at higher speeds. This is probably due to the great mass in combination with the suspension that is designed to make the ride comfortable for the passengers. This strong swinging combined with the fact that the train is considerably higher than a regular car results in a displacement of the camera system that can not be predicted easily. This means that even if two frames are acquired at the exact same position of the vehicle at two different times the position and orientation of the camera with respect to the world coordinate system is not the same.

A final but very significant property are the weather conditions. Trains are operated at constant speed over a much greater range of weather conditions than a normal car. For example, even if the track is nearly fully covered with snow the trains are still operated with no or only a slight reduction in speed, because they need to keep their timetable.

4 Track detection



(a) Track detection using the geometrical properties

(b) Acquisition of the scene and conversion into a birds-eye view orthographic projection

Fig. 2

4.1 Motivation

Obviously a train does not need to do lane keeping but there are two reasons why our autonomous train needs a track detection mechanism:

1. Provide a region of interest to the other sensor systems.
   We are developing a system that uses multi-sensor fusion to achieve reliable and robust decisions. To be able to detect obstacles on or close to the track (which we will call the extended clearance gauge) the system needs to know which part of the sensor data belongs to our physical region of interest. To achieve this the track detector provides the exact position of the detected track in the image to the other sensors (RADAR, LIDAR, Stereoscopic system). Those sensors will be calibrated relative to a point on the train and thus can interpret the track information in their own coordinate system.
2. We assume that most obstacles have a connection to the ground (wheels, feet, ...) and thus would at some point disrupt the appearance of the track. Depending on the camera position, camera angle and distance of the object to the camera those discontinuities in the track are going to appear farther or closer from the train. With sufficient distance from the train those discontinuities get closer to the real distance of the obstacle. Thus this method is best suited for long distance obstacle detection where stereo and LIDAR scanners do not work properly.

4.2 Inverse Perspective Mapping

Based on the observations of the railway track properties we are able to design algorithms that are optimized for those scenarios. A schematic overview of the algorithm is shown in Figure 2a. By transformation of the perspective view into a birds-eye orthogonal view (Inverse Perspective Mapping IPM [4]) we gain the ability to directly check all the geometric constraints that our algorithm requires. In addition, the transformation also makes it easier for appearance based algorithms to find matching regions because the perspective projection does no longer deform the objects depending on their location and thus the track width remains constant over the whole image.

Figure 2b shows the IPM step. The camera acquires the scene in front of the train which is transformed through a perspective projection (Figure 2b *acquired image*). While it is possible to find parallel lines in perspective images [11] it is much simpler if one has the undistorted view. As our algorithm heavily relies on the fact that the tracks are parallel with constant distance at all times, it makes sense to perform an IPM prior to the track detection. We also mentioned that the train undergoes a swinging which translates and rotates the camera in the world coordinate system. To be able to correctly calculate an Inverse Perspective Mapping we use the camera parameters to calculate the perspective projection of every point in the birds eye view. This is slightly more complicated than warping the input image but gives us more flexibility in dealing with the moving camera and non-planar surfaces (which are assumed by image warping techniques).

To calculate the IPM we define a region of interest in world coordinates (for example: 6 meters left, 6 meters right and from 5 to 35 meters in front of the camera). Currently we also require the world in front of the camera to be a planar surface and thus assume a z-Coordinate of zero. This however can be changed in the future once the real curvature becomes available through the LIDAR scanner. The extrinsic and intrinsic camera parameters are calibrated offline because they are not going to change once the system is in place.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} ROI_{left} + \frac{ROI_{width} * x_{ipm}}{IPM_{width}} \\ ROI_{near} + \frac{ROI_{length} * y_{ipm}}{IPM_{height}} \\ 0 \end{pmatrix} \tag{1}$$

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = T_{ext} + R_{ext} * \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{2}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{x' * f_x}{z'} + c_x \\ \frac{y' * f_y}{z'} + c_y \end{pmatrix} \tag{3}$$

We need to scale and offset the points in the IPM image to fit in the desired ROI (as seen in equation 1). ROI_{left} denotes the outermost left point of our (physical) region of interst and ROI_{near} defines the closes point of our region of interest. Combined with the width (ROI_{width}) and height (ROI_{height}) the region of interest is completely defined in physical space (see Figure 2b), because for now we assume the ground to be a flat surface. In equation 2 the extrinsic camera parameters R_{ext} (rotation of the camera) and T_{ext} (translation of the camera) are used to transform the world coordinates of our ROI into the camera coordinate system. Those points are then projected onto the image plane by using the intrinsic camera parameters f_x, f_y focal length and c_x, c_y center of projection (equation 3). This finally establishes a relation between the points in the IPM image (x_{ipm}, y_{ipm}) and the points in the acquired image (u, v). This is done for every pixel in the IPM image and thus reverses the perspective projection of the camera as seen in Figure 2b *perspective correction*.

4.3 Algorithm

Once the IPM image has been created a *Difference of Gaussian* (DoG) filter is applied, which has been proven to be the most versatile in our tests, with rather big kernel sizes (17 for the first filter and 13 for the second) to account for the motion blur and the probably wrong focus of the fix-focus camera. This filtered image is thresholded to create a binary image which is then split vertically into 10 equally sized slices. This breaks up longer lines and allows us to find parallel line segments. It also makes the next step more robust against erroneous edges from the DoG filter.

After this step we got a binary image with blobs that are not higher than $\frac{1}{10}th$ of the image. Those blobs are approximated by fitting a straight line segment to the boundary points. One of the strongest properties of railroad tracks is the constant distance of the tracks and thus the parallel line constraint of short track segments. In the next step the algorithm deletes all candidates that do not have a second candidate within the correct track distance and the correct angle. Those candidates with more than one correct partner are ranked higher than those with fewer partner candidates.

This already creates very good candidates (on average less than 100 track parts where about 20% to 50% do not belong to an actual track). We are using two versions of our algorithm which differ in the last step. The first one selects the correct candidates by recursively combining track candidates from different slices and fitting a 2^{nd} order polynomial to them. Finally two longest combined candidates with the correct distance are chosen as the left and right rails. The second algorithm calculates a center-line for each parallel pair of rail segments and does RANSAC fitting of a 2^{nd} order polynomial to points of the center-lines and the center of the pivot point (P_{zero}) of the leading axle. This property is derived from the zero lateral offset constraint in Table 1 which forces the center-line of the track to always pass through P_{zero} . We also know that the camera is fixed relative to P_{zero} and thus all center-lines have to pass roughly through a single point outside the IPM image.

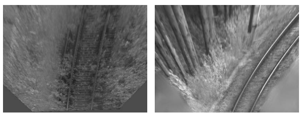
5 Examples

For our tests we used a *Basler scout scA1300-32gm* grayscale camera capable of acquiring 32 frames per second at a resolution of 1296x966 pixels.

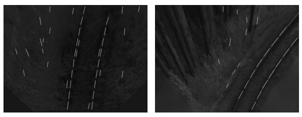
Our tests have shown that this algorithm performs very well even in challenging scenarios. An overview of common scenarios is provided in Figure 1. We can see that the amount of vegetation that is allowed on the track has a large impact (Figure 1c) on the appearance of the space between the tracks. Figure 1e shows one of the best case scenarios for track detection where the top of the rail reflects light quite well. But as already mentioned in Section 3 one can not rely on this feature. Under bad lighting conditions the brightness difference between the tracks and the surrounding area gets problematically low (Figure 1d). This remains a challenging scenario.

5.1 Individual fitting

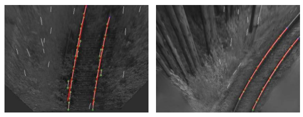
In Figure 3 we can see the output of the first variant which fits the left and right rails individually. The inverse perspective mapped images in Figure 3a represent an area of 8x30 meters starting 12 meters in front of the camera. After the detection of possible track segments and applying the geometric constraint filtering that searches for parallel partner segments with correct distance, we can see (Figure 3b) that most of the remaining candidates do actually belong to the left and right rails. The correct segments are finally determined by fitting the polynomial to various combinations of the candidate segments. Those with the highest score are chosen as shown in Figure 3c.



(a) Inverse perspective mapping of the input images



(b) Detected line segments after parallel line and distance filtering

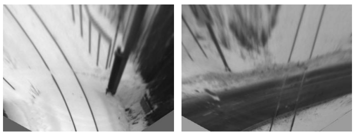


(c) Detected tracks after selecting the best fit for left and right rails

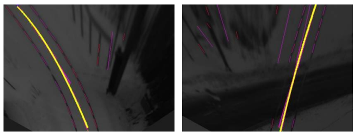
Fig. 3: Examples of fitting individual curves to the two tracks

5.2 Center-line fitting through a pivot point

The individual fitting is supplemented by the fitting of a single curve to the center-line between the left and right rails. In Figure 4b the output of the center-line fitting is shown. The right image in 4a shows a track that is covered with snow and some track candidates that can not be detected because the edges are to weak (Figure 4a right image on the street). But the knowledge that the center-line of the track must pass through the pivot point still allows the algorithm to robustly detect the track.

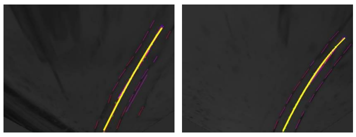


(a) Inverse perspective mapping of the input images



(b) RANSAC fitting of a center-line curve

(c) Challenging Scenarios: input images with raindrops on the windshield



(d) Challenging Scenarios: Still good detection despite rain on the windshield

Fig. 4: Fitting of a single curve to the common center-line of the left and right tracks

One of the core assumptions of this algorithm is the fact that most tracks will not be detected as a whole but rather in parts. So a core task is to find track elements that belong together even if they are not connected. We can see such scenarios where a track could never be detected in one piece in Figure 4c. Those images belong to the worst case scenarios. Although the final system is going to cope with raindrops physically it should still be able to manage those scenarios. We can see in Figure 4d that the current version of the detector is already able to find the correct position of the tracks.

5.3 Reliabilty of the results

Applying the track detection to each frame independently is of course more prone to errors than using the knowledge of previous frames to filter out wrong candidates (i.e. with Kalman filtering). However to show how the algorithm performs on our data we applied the detection on every frame without further knowledge of the context. Every frame was classified by us as either correct or incorrect. The test was applied on a dataset acquired in winter with average weather conditions. The tracklength is approximately 15 kilometers and the train was driving at a speed of about 40 kilometers/hour. The data was recorded at ~ 10 frames per second. The track was incorrectly detected in 297 frames of a total of 13610 recorded frames. This means that the track has been correctly detected 97,81 percent of the time without using temporal constraints.

6 Conclusions

We have presented a novel technique that uses track detection to find the exact location of railroad tracks in an image which can be used by another system to actually detect obstacles along those tracks. The algorithm combines several techniques from lane detection in automotive systems and extends them by applying simple but strong geometric constraints that are provided by the use case (railway) itself. Those geometric constraints allow a reduction of processing cost in the final fitting stage and also generate more robust output even in very challenging scenarios.

However the initial edge detection is rather simple and not designed for the custom rail properties. This will be improved in future versions which should again increase robustness. Additionally the detection will be filtered by using the knowledge of previous frames and the estimated motion of the train to increase robustness.

References

1. H.-Y. Cheng, C.-C. Yu, C.-C. Tseng, K.-C. Fan, J.-N. Hwang, and B.-S. Jeng. Environment classification and hierarchical lane detection for structured and unstructured roads. *IET Computer Vision*, 4(1):37–49, 2010.
2. Radu Danescu and Sergiu Nedevschi. Probabilistic lane tracking in difficult road scenarios using stereovision. *Trans. Intell. Transport. Sys.*, 10(2):272–282, 2009.
3. Oliver Gebauer and Wolfgang Pree. Towards autonomously driving trains. `http://www.softwareresearch.net/fileadmin/src/docs/publications/C086.pdf`, 2008. Workshop for Research on Transportation Cyber-Physical Systems.
4. Fay Huang and Reen-Cheng Wang. Low-level image processing for lane detection and tracking. In *Arts and Technology*, volume 30 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 190 – 197. Springer Berlin Heidelberg, 2010.
5. Gang Yi Jiang, Tae Younf Choi, Suk Kyo Hong, Jae Wook Bae, and Byung Suk Song. Lane and obstacle detection based on fast inverse perspective mapping algorithm. *IEEE International Conference on Systems, Man and Cybernetics*, 4:2969 – 2974, 2000.
6. King Hann Lim, Kah Phooi Seng, Li-Minn Ang, and Siew Wen Chin. Lane detection and Kalman-based linear-parabolic lane tracking. *Intelligent Human-Machine Systems and Cybernetics, International Conference on*, 2:351–354, 2009.
7. Christian Lipski, Bjorn Scholz, Kai Berger, Christian Linz, Timo Stich, and Marcus Magnor. A fast and robust approach to lane marking detection and lane tracking. *Image Analysis and Interpretation, IEEE Southwest Symposium on*, 0:57–60, 2008.
8. Frédéric Maire. Vision based anti-collision system for rail track maintenance vehicles. In *AVSS*, pages 170–175, 2007.
9. Sovira Tan, Jason Dale, Andrew Anderson, and Alan Johnston. Inverse perspective mapping and optic flow: A calibration method and a quantitative analysis. *Image and Vision Computing*, 24(2):153 – 165, 2006.
10. Fraunhofer Institut Verkehrs und Infrastruktursysteme. Hinderniserkennung für Schienenfahrzeuge. `http://www.ivi.fhg.de/frames/german/projects/produktbl/hinderniserkennung.pdf`, 2005.
11. Yue Wang, Eam Khwang Teoh, and Dinggang Shen. Lane detection and tracking using b-snake. *Image Vision Comput.*, 22(4):269–280, 2004.
12. Yong Zhou, Rong Xu, Xiaofeng Hu, and Qingtai Ye. A robust lane detection and tracking method based on computer vision. *Measurement Science and Technology*, 17(4):736, 2006.