Obstacle-Free Range Determination for Rail Track Maintenance Vehicles

Frederic Maire
FaST QUT and NICTA QRL
Brisbane, Australia
f.maire@qut.edu.au

Abbas Bigdeli
NICTA QRL
Brisbane, Australia
Abbas.Bigdeli@nicta.com.au

***Abstract*—Maintenance trains travel in convoy. In Australia, only the first train of the convoy pays attention to the track signalization (the other convoy vehicles simply follow the preceding vehicle). Because of human errors, collisions can happen between the maintenance vehicles. Although an anti-collision system based on a laser distance meter is already in operation, the existing system has a limited range due to the curvature of the tracks. In this paper, we introduce an anti-collision system based on vision. The two main ideas are, (1) to warp the camera image into an image where the rails are parallel through a projective transform, and (2) to track the two rail curves simultaneously by evaluating small parallel segments. The performance of the system is demonstrated on an image dataset.**

***Index Terms*—Computer vision, obtacle detection, train, anti-collision system.**

I. Introduction

Train track maintenance vehicles travel in convoy while moving from one work site to another. The first vehicle of the convoy uses the same signalization system as passenger and freight trains, but the other convoy vehicles follow the preceding vehicle ignoring this signalization system. Because of human errors, collisions can happen between the maintenance vehicles. An anti-collision system based on a laser range finder was introduced to help the track maintenance train operators keep a safe distance between the convoy vehicles. Because of the curvature of rail tracks, the current detection system is reliable only to about 25 meters. NICTA was approached by NSW Railcorp to investigate whether a computer vision system could extend this range for curved tracks. One of the project requirements was that any new solution was not to rely on radio links or GPS devices.

In this paper, we describe a prototype solution implemented in C++ that has been tested on video recordings. The final system will run on an embedded computer in the vehicle. The system induces in real time the curves of the rail track in front of the train, and computes the length of the obstacle-free zone identified.

In the rest of this section we review related work. In Section II, we describe our system. In Section II-A, we explain how the system is calibrated. Section III presents some experimental results.

A. Previous Work

Existing train anti-collision systems like the European Rail Traffic Management System (ERTMS) used in Europe rely on infrastructure equipments. The ERTMS is divided into various levels, as explained in the Strategic Rail Authority Website [1]. Level 1 corresponds to the simplest configuration with fixed blocks and consists of trackside equipment that monitors individual signals and passes this information to the trains via trackmounted transponders. Level 2 is also a fixed block system but a radio link allows a continuous exchange of data between train and trackside, through the GSM-R mobile communication network. This permits the train to reach its maximum permitted speed within its block while maintaining safe braking distances.

More than a decade after autonomous-system technologies emerged in projects such as those from Universitat der Bundeswehr Munich [2] and from the NavLab group at Carnegie Mellon University [3], road lane markings detection systems are mature enough to be commercially available as driver assistance systems [10], [12]. Variations of the Hough transform have been applied for road lane marking detection [13], [14]. In more recent years, pushes for autonomous road vehicles like the DARPA challenges have been playing an important role in the development of new obstacle detection techniques. Active sensors such as lidar and radar are key components of such systems. Most existing vision methods related to obstacle detection studied over the last few years have been for roads. However, there are no operational computer vision systems for obstacle detection on rail tracks in moving trains [5]. Most train related vision systems are used for the monitoring of railroad crossings [6], [7], [8]. In these projects, the monitoring of the rail track area is done with ground based cameras [9].

B. Approaches Explored

In an initial feasibility study, we found that existing techniques for road lane marking detection could be adapted for rail detection for some lighting conditions. However, the metallic surface of a rail does not always behave like a road lane marking. The specular reflection of the metallic rails can make road lane marking detection techniques fail. For example, in sunny conditions a rail looks like a bright band against a darker background (see Figure 1), whereas in a tunnel a rail might look like a dark band against a brighter background (Figure 2).

In [4], we presented an algorithm to induce a model of the track by fitting a piecewise quadratic function to a set of can-

978-1-4244-7815-6/10/$26.00

Fig. 1. An image with glare. The rails are bright.

Fig. 2. An image from a tunnel. The rails are dark.

ditate rail points. Later experiments on a larger dataset showed that the system needed improvement. We then tested a system idea based on an adaptive threshold for Canny edge detection. Although we could cope with gradual changes in lighting conditions, the automatic recalibration for sudden changes was not considered reliable enough (dropping frames). The solution introduced in this paper does not rely on thresholding and can cope with a wide range of lighting conditions. We also tried to track the rail with triplets of similar small boxes; the central box was centered on the rail, and a similar box was searched for just above and below the central box (at the next step the top box becomes the new central box). The system worked fairly well in general, but its sensitivity to patches along the rail made the determination of a rail interruption difficult.

II. System Description

The geometry of rail tracks is much simpler when considered from a bird's-eye view. Indeed, an observer high in the air zooming down on the rail track gets a view of the rails that is not as much distorted by perspective as an observer in the driver cabin of the train. In particular, from above the rail lines appear parallel and the sleepers are spaced regularly.

During a calibration process that is executed only once, our system determines a projective transform Φ that generates a bird's-eye view of the tracks from the image collected in the cabin (see Figures 10 and 11 for examples of warped images).

The gradient image of the warped image enables us to find pairs of parallel segments whose saliency is locally maximum. Because of the lateral motion of the cabin, the horizontal offset (x-axis) of the rails at the bottom of the image can vary dramatically from frame to frame. However, the angle of the rails at the bottom of the image varies more slowy. This fact is exploited to reduce the search space of the rail segments. As Figures 10, 11 and 12 illustrate, our system track the rail curves incrementally by small segments (alternating blue and green segments making the curved ladder in those figures. The thick red segment across the rail track marks the point up to where the system is confident that there are no obstacles on the rail track.

A. Initial Calibration

To compute the projective transformation that maps the image taken by a camera in the driver cabin to a bird's-eye view, we determine the unique projective transformation Φ that maps the four corners T_1, T_2, T_3, T_4 of a trapezoid on the rail track plane to the corners R_1, R_2, R_3, R_4 of a predefined rectangle (See Figure 3). The trapezoid is defined by four lines; the line orthogonal to the rails at 20 meters in front of the camera, the parallel line at 50 meters, and the two straight rail lines. In Figure 4, the red line corresponds to the $20m$ line, and the blue line correponds to the $50m$ line. In our system, a human operator positions on the screen these two lines by moving them up and down with the keyboard. The green line corresponds to the horizon line. The horizon line is easily computed from the $20m$ and $50m$ lines. Once the trapezoid is specified, we compute the unique projective transformation Φ satisfying $\Phi(T_i) = R_i$ for $i \in [1, 4]$, where the predefined rectangle is positioned centrally at the bottom of the image. This rectangle is of width 60 pixels and of height a quarter of the camera image. With this calibration process, the image seen by the virtual observer covers the area in front of the train from the line at 20 meters to the line at $140 = 20+4\times(50-20)$ meters.

The rail lines are determined with a Hough transform. The Hough transform requires a binary edge image. As the images considered are outdoors images, the lighting conditions can vary dramatically. Suitable thresholds for a Canny edge detector depends heavily on the lighting conditions. We did not want to have to rely on the human operator to select the values of the high threshold θ_H of the Canny edge detector. However, we ask the operator to draw a rectangle in a region of the screen where there is a rail. To find a suitable value for θ_H, we initialize it with a large constant, and reduce this value until the number of edge points is large enough so that a line segment as long as the rectangle drawn by the operator is

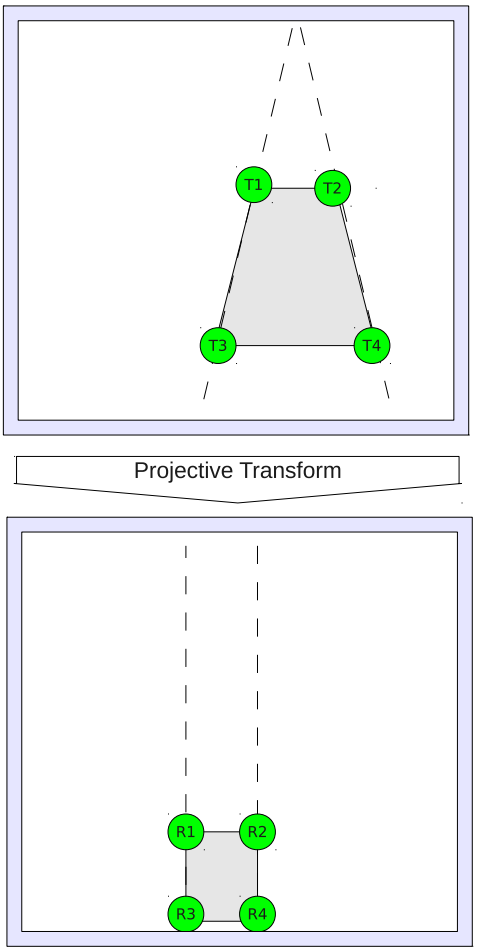

Fig. 3. Projective transform mapping the corners of the trapezoid of the cabin view to the corners of the rectange of the bird's-eye view.

detected (see Figure 7). The line segment should correspond to a rail.

Figure 5 shows a typical Canny edge image produced by the semi-automatically selected θ_H threshold. The Canny image contains enough points to identify the rail lines, but the number of points is small enough that it is very likely that the rail lines will be among the first 20 lines returned by the Hough transform. Once θ_H is computed, the Hough transform is applied. The operator is then asked to click the mouse between the rails of interest. The left and right Hough lines closest to the point where the operator clicked the mouse should correspond to the left and right rails. The resulting lines are shown in red in Figure 6.

The detection of the start of the rails at the bottom of the image and the incremental building of the rail curves are based on the same function. This key function is described in the next sub-section.

B. Groping The Rails

The critical element of our system is a function grope_rails(R, α_e, g_e) that given a region of interest (ROI)

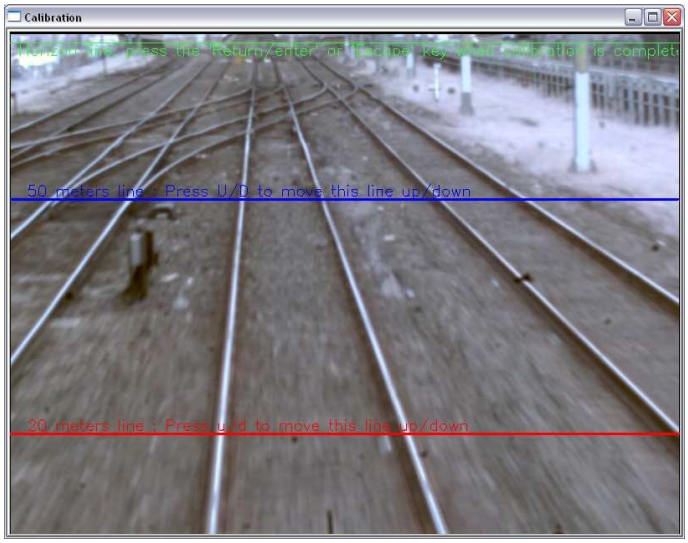

Fig. 4. The calibration trapezoid is specified by the lines at $20m$ and $50m$ and a pair of rails.

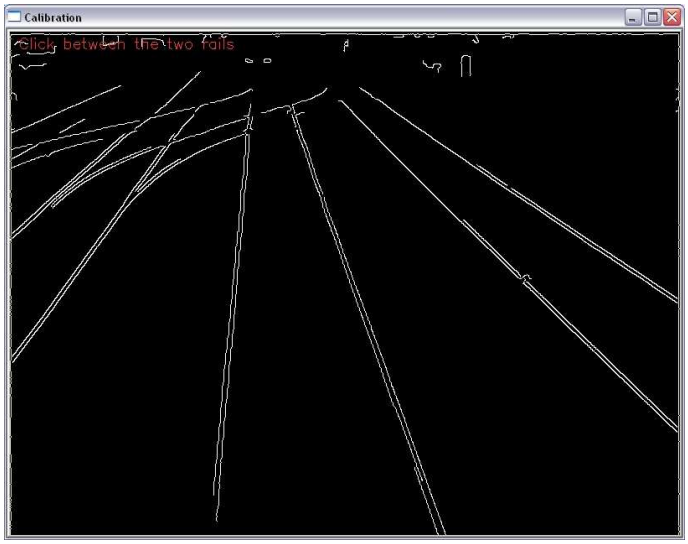

Fig. 5. The Canny edge detection threshold parameters are selected automatically.

delimited by the rectangle R, an expected angle α_e and an expected gap g_e, returns the pair of parallel segments maximizing some score function proportional to the posterior probability that these segments correspond to rail segments (see Figure 8). To avoid the thresholding issues mentioned in Section I, we rely on the relative gradient magnitudes of the horizontal derivatives of the pixel intensity. The average intensity $M_{i,j}$ of the horizontal gradient of the image along the segment joining a point T_i at the top of the rectangle R to a point B_j at the bottom of R is stored in a matrix M. The segment $[T_i, B_j]$ shifted horizontally by g pixels is $[T_{i+g}, B_{j+g}]$. The score of the pair of parallel segments $([T_i, B_j], [T_{i+g}, B_{j+g}])$ is $M_{i,j} \times M_{i+g,j+g} \times \exp\left(\frac{-|g - g_e|^2}{2\sigma^2}\right)$. This formula follows a standard Bayesian

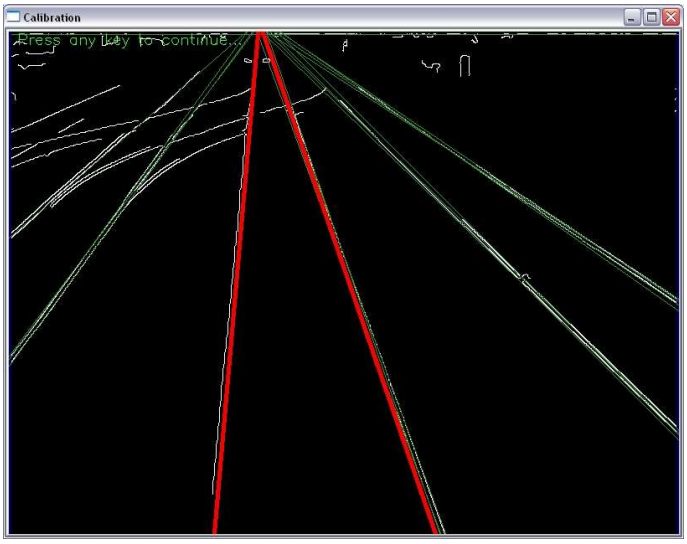

Fig. 6. The rail lines are detected by a Hough transform. The operator clicked in the Hough image between the rails of interest. The identified pair of rails is highlighted in red.

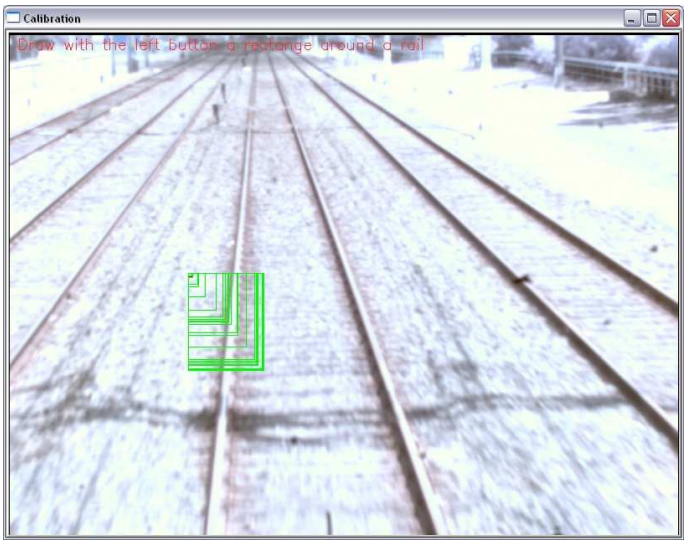

Fig. 7. A rectangle choson by the human operator to help automatically select the Canny threshold.

framework $\texttt{posterior} \propto \texttt{likelihood} \times \texttt{prior}$. The factors $M_{i,j}$ are related to the likelihood $P(\text{gradient} \mid \text{rail})$. The exponential factor reflects our prior knowledge on the gap between the rails.

That is, a pair of segments is penalized if it has a gap g different from the expected gap g_e. For efficiency, only relevant entries of the matrix M are computed; tolerance constants are used for the angle of the rail segment and the expected location of the bottom position B_j of the rail segment with respect to the ROI. The dotted rectangles in Figure 8 represent two regions of interest.

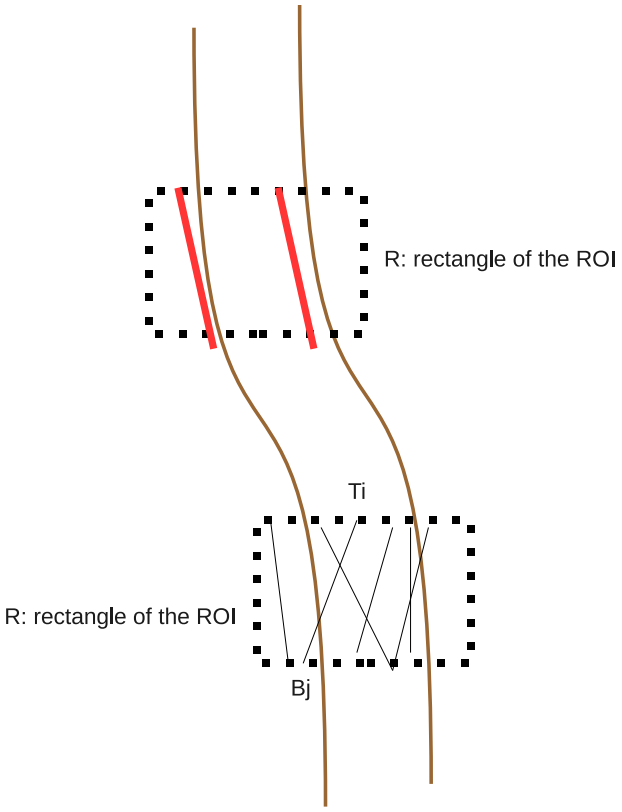

Fig. 8. Search for parallel segments (in red) is performed in small rectangular ROI's.

C. Rail Starts Localization

To locate the starts of the rails, the search rectangle R is set to the whole width w of the non-black bottom of the image (see Figure 9). If we did not have the constraints on the rail angle, the complexity of the search would be approximately in $O(w^2)$. However, the angle constraints keeps the complexity in $O(w)$. Similar remarks apply for the subsequent segments searches.

One of the previous systems we experimented with was considering only one rail at a time. This system could sometimes latch on white bands on the quay. The fact that the two rails are considered simultaneously prevent this problem from happening in the present system.

D. Obstacle Detection

At each step of the curved ladder, the direction of the edge gradients is compared with the angle of the pair of segments returned by the function grope_rails(R, α_e, g_e). If they are consistent (edge gradients orthogonal to the segments returned), the system considered that it is still tracking the

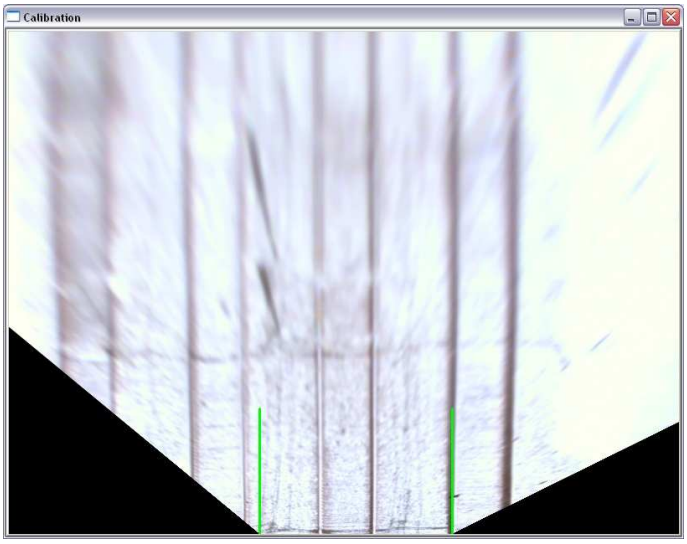

Fig. 9. The search region for the starts fo the rails is delimited by the two green vertical segments.

rails. Otherwise, the system declared that it is not confident that there is no obstacle at that point.

III. Experimental Results

The system runs in real time (more than 25 frames per second on a laptop). We have tested the system on an image dataset collected by a camera installed in a train cabin. The images were taken with different camera settings (tilt, zoom). Because there was no calibration done at the time of the recording (specification of the 20m and 50m lines), we can only set those lines by guesswork. On the images we were provided with, the vision algorithm detects reliably the obstacle-free zone to more than 100 metres. Some videos demonstrating the robustness of the system are available at
http://www.youtube.com/watch?v=7vzJ5TObZn0
http://www.youtube.com/watch?v=EK4AnCpagKk
http://www.youtube.com/watch?v=1Akr3SEKXz4

If there is a fork in the rail track, although the system appears biased towards the correct path, it will pick rather randomly one of the two paths. This is a limitation of the present system.

IV. Conclusion

We have designed and implemented in C++ (using the OpenCV computer vision library) a prototype anti-collision system for rail track maintenance vehicles that monitors the space ahead of the train for obstacles. This system can help maintain a safe distance between maintenance trains. The system features an original semi-automatic calibration module that fully exploits the constrained geometry of rail tracks. We believe that the strength of our system is due to two main reasons;

1) our approach avoids the problem of the selection of Canny edge thresholds (by using the gradient images directly),
2) the projective transformation provides an image where the rails are parallel (making the simultaneous tracking of the two rails easy).

The system shows robustness under a wide range of illumination conditions. We expect better performance on the embedded system as it will use a better quality camera. We have no data for rain and fog, but testing under these weather conditions will be performed with the embedded system.

Acknowledgment

The first author would like to thank Ms Irina Gordienko for her invaluable help. This study was conducted under contract with NSW Railcorp, and the authors would like to thank the company personnel for their cooperation.

References

[1] SRA, 2004. European rail traffic management system. Website http://www.sra.gov.uk
[2] E.D. Dickmanns and A. Zapp, *Autonomous High Speed Road Vehicle Guidance by Computer Vision*, Automatic Control World Congress, 1987: Selected Papers from the 10th Triennial World Congress of the Intl Federation of Automatic Control, Pergamon, 1987, pp. 221226.
[3] C. Thorpe, *Vision and Navigation: The Carnegie Mellon NavLab*, Kluwer Academic Publishers, 1990.
[4] Maire, F.; , *Vision based anti-collision system for rail track maintenance vehicles*, Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on , vol., no., pp.170-175, 5-7 Sept. 2007
[5] Kaleli, F.; Akgul, Y.S.; , *Vision-based railroad track extraction using dynamic programming*, Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on , vol., no., pp.1-6, 4-7 Oct. 2009
[6] Yaser Sheikh, Yun Zhai, Khurram Shaque, and Mubarak Shah. *Visual monitoring of railroad crossings*, 2004.
[7] J. Vazquez, M. Mazo, J.L. Lazaro, C.A. Luna, J. Urena, J.J. Garcia, J. Cabello, and L. Hierrezuelo. Detection of moving objects in railway using vision. pages 872875, 2004.
[8] Jun Xue, Jun Cheng, Li Wang, and Xiaorong Gao, *Visual monitoring-based railway grade crossing surveillance system*, Congress on Image and Signal Processing,2:427431, 2008.
[9] Nikolaus Mohler, Milan Ruder and Faruque Ahmed, *An obstacle detection system for automated trains*, pages 180185, June 2003.
[10] Ieng S-S, Tarel J-P. and Labayrade R., *On the Design of a Single Lane-Markings Detector Regardless the On-board Camera's Position*, in Proceedings of IEEE Intelligent Vehicle Symposium (IV'2003),June 9-11 2003, Colombus, OH, USA.
[11] Ieng S-S, *Methodes robustes pour la detection et le suivi des marquages*, These de doctorat de luniversite de Paris 6, Novembre 2004.
[12] Apostoloff, N. and Zelinsky A., *Vision In and Out of Vehicles: Integrated Driver and Road Scene Monitoring*, The International Journal of Robotic Research, Vol 23, No 4-5, April-May 2004, pp 513-538.
[13] B. Yu, A. Jain, *Lane Boundary Detection Using a Multiresolution Hough Transform*, icip, p. 748, 1997 International Conference on Image Processing (ICIP'97) - Volume 2, 1997.
[14] Yue Wang, Eam Khwang Teoh, and Dinggang Shen, *Lane detection and tracking using B-snake*, Image and Vision Computing, 22(4):269-280, April 2004.

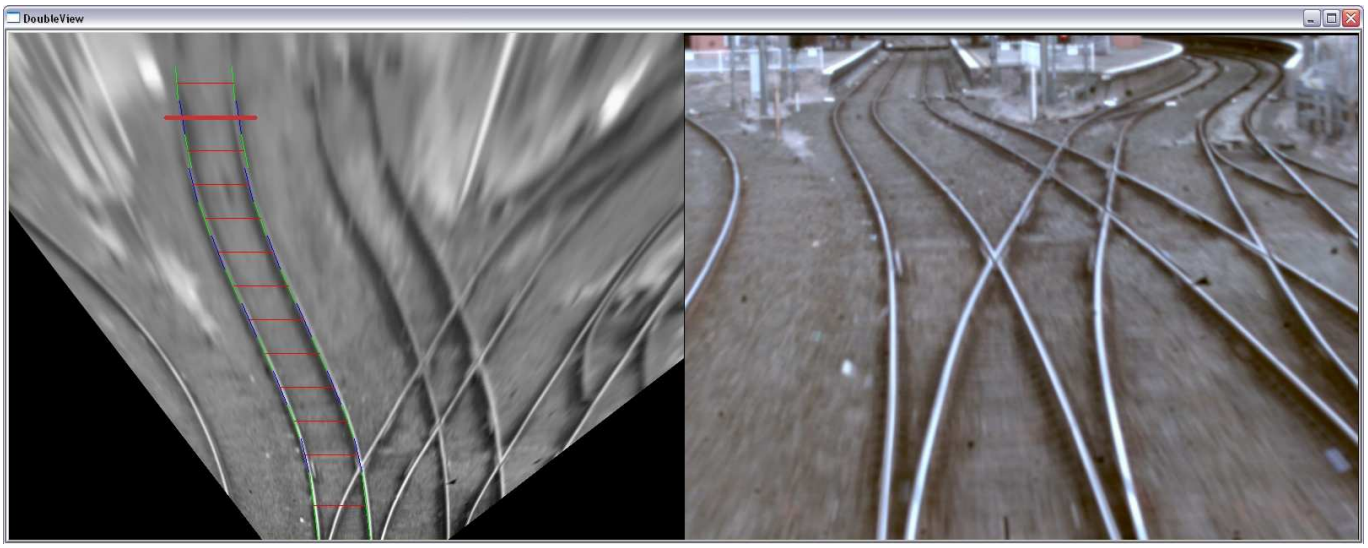

Fig. 10. Frame 2062; although the system follows correctly the rails, it considers the top green segments too weak and stops on the previous segments (in blue).

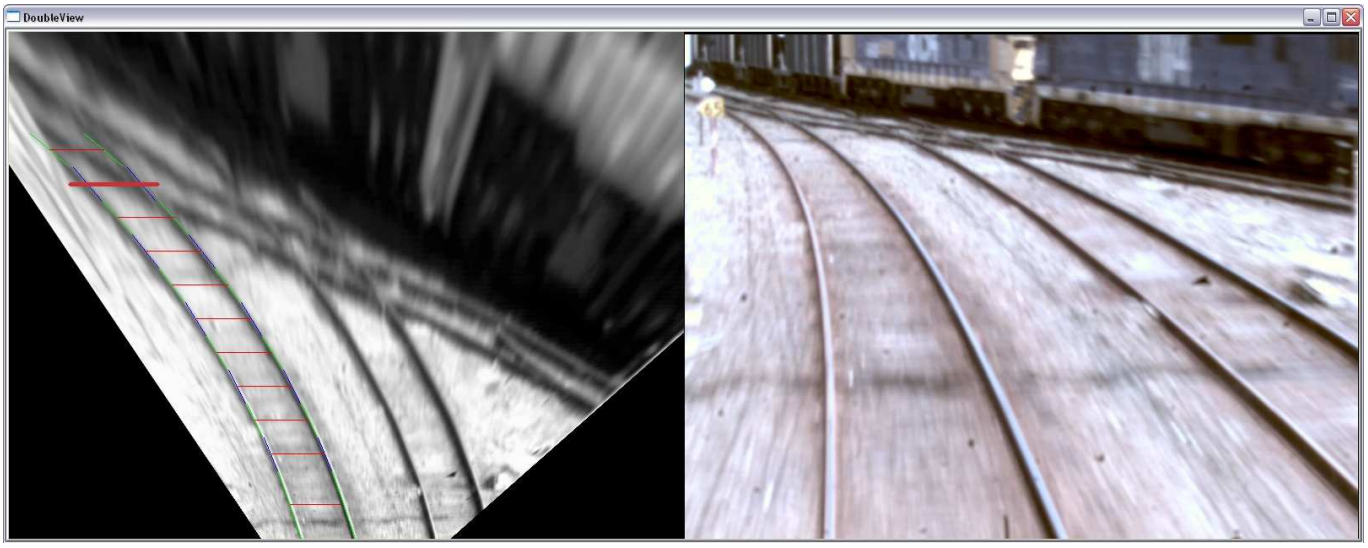

Fig. 11. Frame 3194; the edge gradients in the top green segments are inconsistent with the angle of the segment, therefore the system stops before.

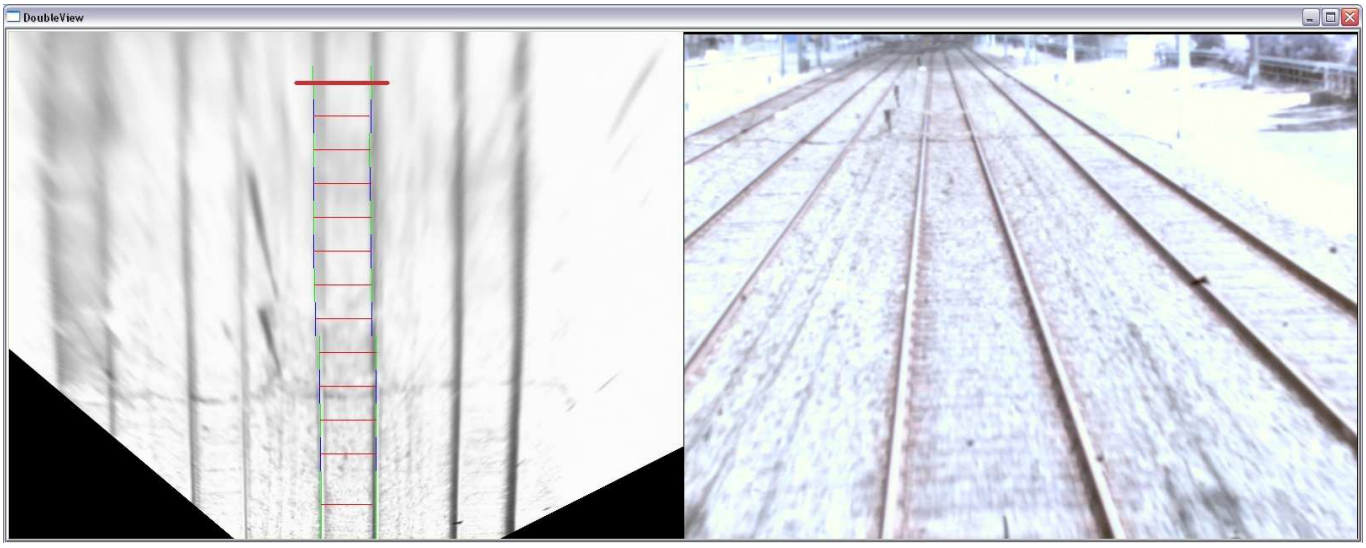

Fig. 12. Frame 1430; Even with an image of poor quality, the rails are tracked properly.