Vision-Based Railroad Track Extraction Using Dynamic Programming

Fatih Kaleli and Yusuf Sinan Akgul
GIT Vision Lab, http://vision.gyte.edu.tr/
Department of Computer Engineering, Gebze Institute of Technology,
Gebze, Kocaeli 41400 Turkey
{fkaleli,akgul}@bilmuh.gyte.edu.tr

***Abstract*—Most of the common driver assistant systems for detection of obstacles work on unstructured environments. These environments generally include many non-planar surfaces which pose a big challenge for vision systems. Similar problems exist for railroad environments which often contain complex shapes and surfaces like hills and vegetation along railroad tracks. In railroad transportation, the main task of a train driver is to carefully focus on the track. Therefore the field of view of a train driver must contain the space between two rails in front of the train and the near lateral area (left and right side) of these rails. In this paper, we present an algorithm to extract the train course and railroad track space in front of the train using dynamic programming in railroad environments. We use dynamic programming to compute the optimal path which gives the minimum cost to extract the railroad track space. The proposed algorithm extracts the left and right rails using dynamic programming simultaneously. Our method does not need any static calibration process. For this purpose, a camera system was installed in front of a locomotive. Experimental results show the effectiveness of the algorithm.**

I. Introduction

In recent years, driver assistance systems have been playing an important role to improve the safety on tracks. Obstacle detection is one of the main objectives of these systems. The majority of existing vision methods related to obstacle detection have been extensively studied over the past several years on normal tracks (e.g. highways, country roads). However, there are limited number of computer vision systems for the railroad obstacle detection. The main task of a train driver is to carefully drive the train according to a route timetable [1]. When we consider that the train drivers work under high stress and worse sleep quality, a special care is needed in railroad operations such as the monitoring of the track in front of the train [1][9]. For this important operation, the field of view of a train driver must contain the space between two rails in front of the locomotive and the near lateral area of these rails. A driver assistant system which employs vision based sensors will provide great benefits to the train drivers to focus on the track.

General vision related studies for railroad applications have been proposed for track inspection systems [2][7][15]. In [10], a sign detection and classification algorithm was studied to increase safety. Most of the stereo-vision based methods have been implemented for the monitoring of railroad grade crossings recently [14][17][18]. These studies are about the monitoring of the railroad track area with fixed station camera systems. Even though these techniques are important solution ways for monitoring of crossings, implementation of these systems in all grade crossings over the railroad networks is not cost effective. On the other hand, the collisions or railroad incidents do not occur in only grade crossings. They may happen at any point along the railroad tracks. Therefore, onboard vision control systems for detection of obstacles must be taken into consideration. In the studies [4][11][12], vision cameras (monocular, stereo etc) have been installed in front of the train engines to reduce the risk of a collision. These implementations can be used in two different types of applications: train driver assistant systems and fully automated train systems [11]. As an example of implementation of board camera systems, [4] presents ego motion estimation of the camera and moving object detection in railroad crossing: it consists of a monocular camera method applied to consecutive frames. Ego motion of the camera is estimated using image measurements of fixed points in the scene. However, railroad environments often contain complex shapes and different surfaces like hills and vegetation along railroad tracks. For this reason, this method does not provide accurate estimation and produces false detections in most cases.

Most of the common driver assistant systems for detection of obstacles work on unstructured environments. These environments generally include many non-planar surfaces which pose a big challenge for vision systems. Similar problems exist for railroad environments. In the study [12], active sensors such as lidar and radar, and camera systems have been used to detect the obstacles on railroad tracks. Another similar application [11] consists of stereo-based vision system to extract the tracks using clothoid model and search the objects on the track by using different well-known methods such as optical flow, stereo based matching by inverse perspective mapping etc. The method [11] does not work on the track curves and far distances. In many implementations, small movements of the camera can affect consecutive images negatively and track detection will not give the accurate results due to the fact that the cameras are mounted on a moving platform. In [13], the real-time stabilization algorithm has been proposed for the video sequences which have been taken from a camera mounted on a locomotive of a high-speed train.

In this study, we restrict ourselves to just extracting the track area from the images which have been captured by a camera system in front of the locomotive. We consider the track area extraction as the most important pre-request step for many obstacle detection methods. We believe that the analysis of the train course will supply important cues to the train driver assistant systems. For this reason, this paper presents an algorithm to extract the train course and railroad track space in front of the train using dynamic programming in railroad environments. Any obstacle detection process based on motion or stereo methods will not be considered in this study.

Dynamic programming is mainly used to tackle problems that exhibit optimal substructure according to proper recursive equations. The solution to a given problem can be obtained by the combination of optimal solutions to its subproblems. Especially, dynamic programming can be used to extract the surfaces which have similar characteristics in images. In our track extraction problem, the characteristics of the railroad environment is appropriate to apply dynamic programming. We use dynamic programming to compute the optimal path which gives the minimum cost to extract the railroad track space. In a related work, [6] presents a method to segment the road lane boundaries by using dynamic programming. Kubota [5] presented a stereo-vision obstacle detection method based on dynamic programming that uses disparity values. Badino [3] presented another dynamic programming based approach to compute the free space in front of the vehicle by using stochastic occupancy grids. Different features such as intensity, color, texture, image gradients, disparity or stochastic data can be used to accomplish the goal of extracting the railroad track space. Accurate computation of the railroad track space requires strong parameters especially on the horizontal lines that the track width reaches minimum values in the image. In this study, our method takes the gradient features from input images to compute the track space.

Main advantages of the proposed algorithm is extraction of the left and right rails using dynamic programming simultaneously. For the computation of the track space, we calculate the initial track rail width for each image of the sequence automatically. For this reason, the proposed method does not need any static calibration process while most of the vision methods need camera calibration. Finally, the proposed algorithm is not affected by camera movement, specifically camera pitch oscillation.

The rest of this paper is organized as follows. Section 2 describes our railroad track extraction algorithm and railroad feature extraction. We show experimental results in section 3. Finally, section 4 gives the conclusions.

II. Railroad Track Extraction

In this section, we present the process of the railroad track extraction. The proposed method consists of the following elaboration steps:

1) Railroad Feature Extraction
2) Track Extraction using Dynamic Programming

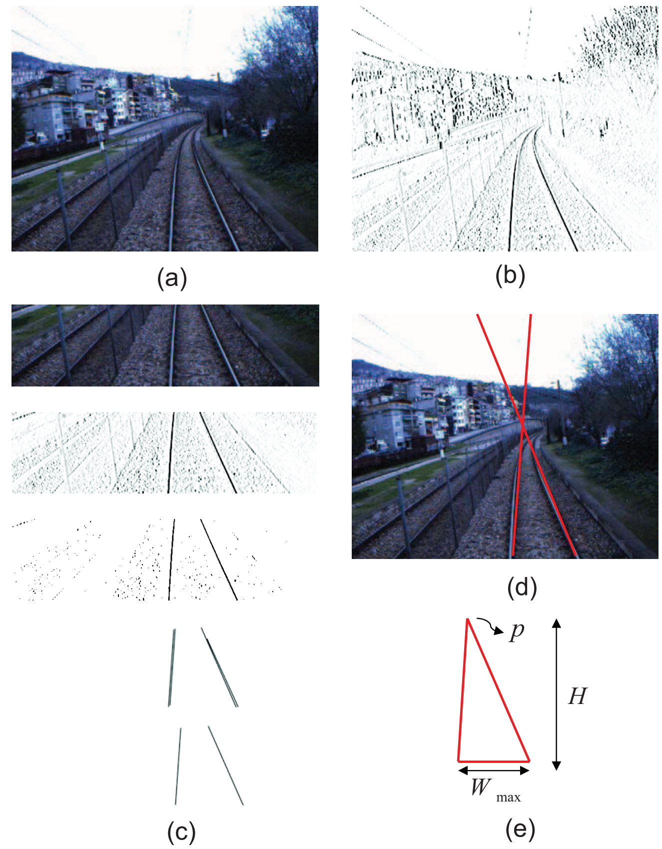


Fig. 1. Example of feature extraction process: (a) original image, (b) gradient image, (c) From top to bottom, input image, gradient image, after threshold, line fitting with Hough transform, rail extraction respectively, (d) line intersection, (e) vanishing point p and W_{max}.

A. Railroad Feature Extraction

This section explains how to compute the railroad vanishing point p and track width parameter W_{max} from an input image. Rail tracks are aligned parallel to each other and their width decrease linearly in the image from bottom to up as it can be seen in Fig.1(a). We use the vanishing point term to refer to the intersection of the track lines defined as in Liang [8]. The height H is the distance from the bottom row of the input image to vanishing point p of the track rails and the track width parameter W_{max} is the distance in pixels between the track rails (see Fig.1(e)). These features are used to extract the track space in front of the train.

For computation of the railroad vanishing point and track width parameter W_{max}, we propose a process for extracting of the rail lines from images. To extract rail lines, we first begin to compute the gradient of the input image by using the Sobel operator. An example of a railroad environment is shown in Fig.1(a) and the gradient magnitude image $G(i,j)$ is displayed in Fig.1(b). (i and j represent pixel locations in the image.) Secondly, a given threshold value is applied to the bottom 1/3 of the gradient image because of the fact that this part of the image contains sufficient information to extract the features.

Moreover, rails are straight and their gradient magnitudes are high in the bottom of the image. Figure 1(c)(first and second rows) show the cropped input image and its gradient magnitude image. This thresholding step eliminates the points which have lower gradient magnitude from the threshold value T. Then a binary image $B(i,j)$ is created from gradient image $G(i,j)$ by the Equation 1. Figure 1(c) (third row) shows the binary image.

$$B(i,j) = \begin{cases} 1 & \text{if } G(i,j) > \text{T} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Next, a Hough transform process is applied to the binary image $B(i,j)$ for the detection of rail lines. This process also removes the noise in the binary image. The result of this effect is shown in Fig.1(c)(fourth row).

Hough transform is very useful for detecting straight lines. For each pixel and its neighborhood, the Hough transform determines if there is enough evidence of an line at that pixel. Thus, it returns many lines which have different lengths. Lines returned contain their end-point coordinates (x,y) without any length information. The length of a straight line between its two end-points (x_1,y_1) and (x_2,y_2) is calculated by Equation 4.

$$l = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2} \tag{2}$$

This process detects many lines accompanying the track rail lines, particularly in double track railway environments. However, we would like to know only two track rails in front of the train. Therefore we extract the rail lines which are the longest lines on the left and right to calculate more accurate vanishing point according to a track rail filter. This filter finds appropriate rail lines in front of the train by the Equation 3. The result of applying this filter to the Hough image $H(i,j)$ is displayed in Fig.1(c)(fifth row).

Let L_1 with its end points $(a,b)(c,d)$ and L_2 with its end points $(e,f)(g,h)$ be the lines which are found at end of the Hough transform process.

$$Lines = \begin{cases} rails & \text{if } |a-e| < DM \text{ and } |c-g| > Dm \\ not & \text{otherwise} \end{cases} \tag{3}$$

where DM is the maximum width value between bottom x coordinate values of two distinct lines and Dm is the minimum width value between up x coordinate values of two distinct lines. This determines the rails on the left and right. Then we can easily find the longest lines on the left and right by computing their lengths by Equation 4. To illustrate this selection, we refer to the example shown in Fig.2.

After we obtain the track rail lines using Hough transform and track rail filter, we can directly compute the intersection of two straight rails in the image. For the computation of the vanishing point, we use the common form of a linear equation. This equation can be described as the following "slope-intercept" form.

$$y = mx + c \tag{4}$$

where m is the slope and c is the y-intercept of the line.

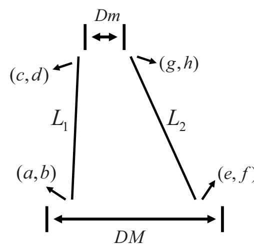


Fig. 2. Track rail filter

The slope equation of a line that passes through two distinct points $P_1(x_1,y_1)$ and $P_2(x_2,y_2)$, is given by 5:

$$m = (y_2-y_1)/(x_2-x_1) \tag{5}$$

The y intersect of a line that passes through two distinct points $P_1(x_1,y_1)$ and $P_2(x_2,y_2)$, is given by 6:

$$c = (x_2y_1 - x_1y_2)/(x_2-x_1) \tag{6}$$

Finally, y coordinate of railroad vanishing point p is calculated by the following Equation 7:

$$p_y = (m_1c_2 - m_2c_1)/(m_1-m_2) \tag{7}$$

where p_y is the y coordinate of the intersection of the lines. We can easily compute W_{max} by subtracting x coordinates of the rail lines at the bottom of the image. Figure 1(d) shows the intersection of the lines (vanishing point p) in the input image.

B. Track Extraction using Dynamic Programming

Dynamic programming is used to compute the optimal path which gives the minimum cost to extract the railroad track space. Computations are done from bottom row to railroad vanishing point p of the gradient image.

The proposed algorithm extracts the left and right rails using dynamic programming simultaneously. It is assumed that the points on left rail are dependent to the points on the right. When the minimum path is calculated by using dynamic programming, the points on the left rail must be considered with the right points according to proper width value. Otherwise, the track extraction step does not work on curves successfully.

To solve the track extraction problem by dynamic programming, firstly the minimum cost of each point of the gradient image is computed according to a graph structure as shown in Figure 4. For these computations, a cost matrix C which stores the cost values is generated. Secondly, the shortest path is calculated using cost matrix C. Finally, each point set which includes the points on left and right rails at each row is extracted from the graph. Figure 3 displays the track extraction step using dynamic programming.

Let a graph $G(V,E)$ represent the input image $I(x,y)$ and (x,y) denote the pixel locations in the image. Each pixel is assigned to each node in graph G. V is the set of nodes and E is the set of edges which are the connections between nodes.

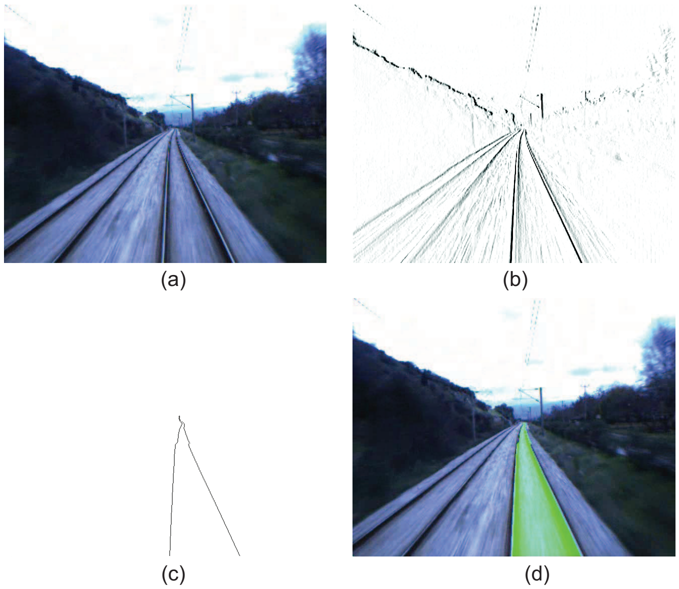


Fig. 3. Example of track extraction using dynamic programming: (a) original image, (b) gradient image, (c) extracted track rails (d) color representation of the track path

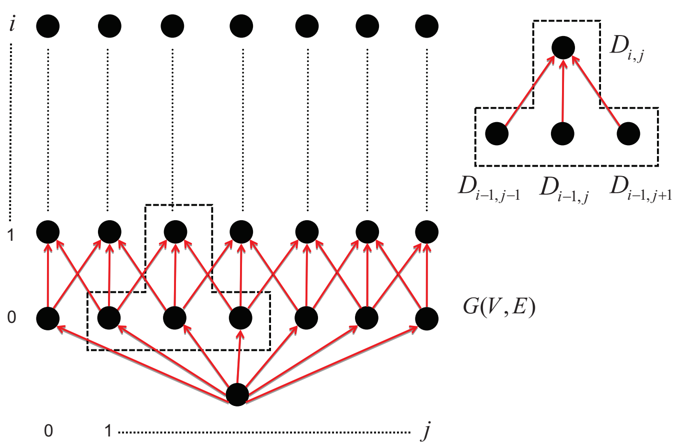


Fig. 4. Graph view: Dynamic Programming is applied to the image from bottom row to railroad vanishing point and each node is associated with a pixel.

In this study, a central node (for example: $D_{i,j}$) is connected to three other nodes in the graph. Fig.4 shows the structure of the graph $G(V, E)$ and connections among nodes.

The proposed algorithm calculates the shortest path using the following recursive equations;

$$P_i(D_{i,j}) = \min_{\substack{\text{all ternary}\\ \text{paths}}} \{C_{i,j} + P_{i-1}(D_{i-1,j})\} \tag{8}$$

$$P_0(D_{0,j}) = C_{0,j} \tag{9}$$

where $P_i(D_{i,j})$ is the optimal path to node $D_{i,j}$ of graph G at row i, and $C_{i,j}$ is the edge cost from the nodes of the lower row to node $D_{i,j}$. The cost-function $C_{i,j}$ returns a cost associated with the node $D_{i,j}$ from the cost matrix C.

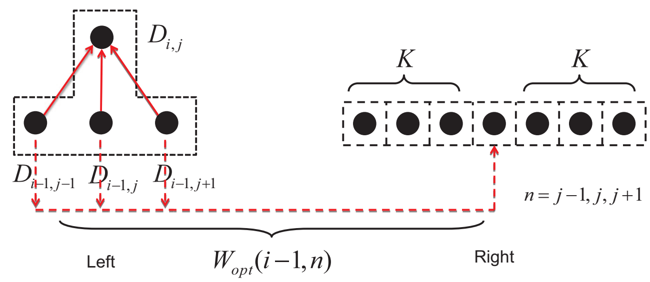


Fig. 5. Left and Right rail

The cost of each edge, $C_{i,j}$, is calculated using the following equations:

$$C_{i,j} = \min_{n \in \{j-1, j, j+1\}} \{C_L + G_L + G_R + W_{LR}\} \tag{10}$$

$$\begin{aligned} C_L &= C_{i-1,n} & (11)\\ G_L &= G(i-1, n) & (12)\\ G_R &= G(i-1, n + M_{i-1,n}) & (13)\\ W_{LR} &= W_{i-1,n}, & (14) \end{aligned}$$

where C_L is the cost function value for node $D_{i-1,n}$, G_L and G_R are the gradient values of the nodes $D_{i-1,n}$ and its right node with row coordinate $n + M_{i-1,n}$ respectively, n is the column index for the nodes $\{D_{i-1,j-1}, D_{i-1,j}, D_{i-1,j+1}\}$ which are connected to node $D_{i,j}$ in graph G, W_{LR} is the width energy term (see Equation 16), and $M_{i-1,n}$ is the distance in pixel from node $D_{i-1,n}$ to its right node which is calculated by Equation 16.

For example, in graph G, let $\{C_B, C_C, C_D\}$ be the edge costs of $\{B, C, D\}$ nodes which are connected to the node A respectively. Edge cost of the node A is equal to the minimum cost to get from the node of the set $\{B, C, D\}$ to it because of the fact that $\{B, C, D\}$ are the only nodes that they can reach it. Therefore, the minimum cost value from the set $\{C_B, C_C, C_D\}$ is assigned to the cost of the edge for the node A, and the the parameter W_{opt} for node A is taken from the node which has the smallest cost.

The width value between track rails plays an important role for the extraction of the path. If a set of consecutive points composes the left rail, there must be another set of points which corresponds to the first set according to a proper width parameter W_{opt} on the right. Fig.5 illustrates left and right rail nodes. As it can be seen in this image, W_{opt} values of left rail nodes $\{D_{i-1,j-1}, D_{i-1,j}, D_{i-1,j+1}\}$ determines the position of the right node in the graph G. In the beginning (for $i = 0$), W_{opt} value of all nodes is equal to W_{max} which is the track width parameter computed in feature extraction step.

The width energy term $W_{i,j}$ is computed from the Equation 16:

$$S = (W_{opt}(i, j) - M_{i,j}) * c \tag{15}$$

$$W_{i,j} = \begin{cases} M_{i,j} + S & \text{if } W_{opt}(i, j) < I_w \\ t & \text{otherwise} \end{cases} \tag{16}$$

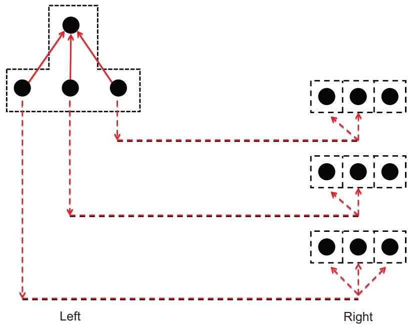
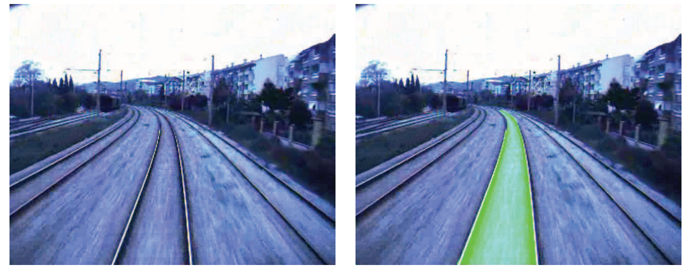


Fig. 6. Usage of neighborhood parameter K rail



(a) (b)

Fig. 7. Examples of railroad track extraction on various scenes: a) Original image b) Result of railroad track extraction

where $W_{opt}(i,j)$ is the width value of node $D_{i-1,n}$ and its initial value is equal to W_{max}. $M_{i,j}$ is the distance value from node $D_{i,j}$ to node $D_{i-1,n+(M_{i-1,n})}$ which has the minimum gradient value on the right according to a neighborhood parameter K, I_w is the image width and S is the smoothing value for the right rail. For computation of S, c is used as a small constant for smoothing the points on the right rail, and t is used as a large constant parameter which penalizes the points if there does not exist any corresponding points to them on the right. Fig.5 represents the calculation of the width energy term $W_{i,j}$.

The use of the neighborhood parameter K is crucial because of the fact that selection of the right node is dependent to left node. When we consider the track width decreases linearly in the image from bottom to vanishing point, determination of the right node must be appropriate to this characteristic of real world railroad tracks. For example, let K be equal to 1. As it is shown in Fig.6, the right node for node $D_{i-1,j-1}$ can be the node which has the distance $W_{opt}(i,j)$ from node $D_{i-1,j-1}$ or the left one. Similarly, the right nodes for other node $D_{i-1,j}$ and $D_{i-1,j+1}$ can be estimated according to principal as it can be shown in Fig.6.

III. Experimental Results

We have applied the dynamic programming based method to many video sequences taken from a camera places in front of a train engine. The proposed algorithm gives good extraction rate during the video sequence under various unstructured environment, especially it extracts the curve of the track successfully in most cases as shown in Fig.7 and Fig.8. Figure 11 shows another example of railroad scene. In the figures, railroad track area is marked out by color.

The quality of the railroad track extraction depends on the parameters (vanishing point and W_{max}) and gradient magnitudes. Experimental result shows that initial track width parameter W_{max} is calculated accurately.

Test data has been taken from the camera system which was mounted on a moving platform when the train was moving under high speeds. The proposed algorithm is not affected by camera movement, specifically camera pitch oscillation. Fig.8 and Fig.9 are the frames of the same video data. As it can be seen in Fig.8(a), the train is not moving. Fig.9(a) has been taken when the train was moving. In each of these images, the track area has been extracted accurately. In Fig.10, some consecutive frames of a different video are displayed. As shown in these frames, our method detects the railroad track space even at high distance.

(a) (b)

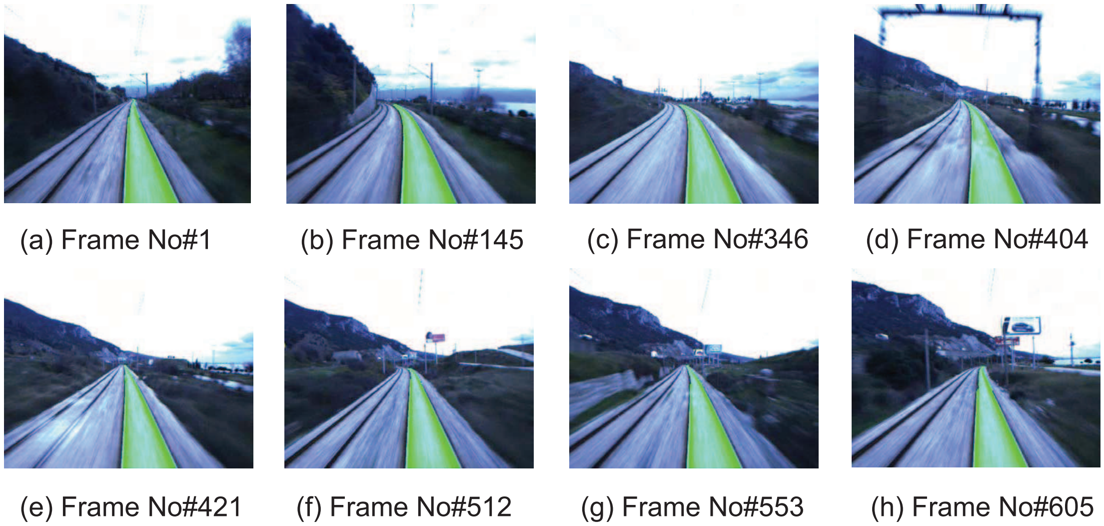
Fig. 8. Examples of railroad track extraction on various scenes: a) Original image b) Result of railroad track extraction

(a) (b)

Fig. 9. Examples of railroad track extraction on various scenes: a) Original image b) Result of railroad track extraction

IV. Conclusions

A dynamic programming based railroad track extraction algorithm has been proposed. The developed algorithm allows extraction of the train course and railroad track space in a robust way from the images captured by an on board camera system in front of the train. The method shows robustness to negative effects of the camera movement when train is running at high speeds along the railway. We consider this



(a) Frame No#1 (b) Frame No#145 (c) Frame No#346 (d) Frame No#404

(e) Frame No#421 (f) Frame No#512 (g) Frame No#553 (h) Frame No#605

Fig. 10. Examples of railroad track extraction on various scenes

(a) (b)

Fig. 11. Examples of railroad track extraction on various scenes: a) Original image b) Result of railroad track extraction

feature of the proposed algorithm may provide some important contributions to estimate ego motion of the camera and its dynamic calibration at the time acquisition.

In future, we aim to develop our algorithm to give more accurate extraction considering railway track clothoid model because of the fact that the rail tracks are built after a clothoid geometry. We believe that the analysis of the train railway model will supply significant information to the train driver assistant systems about the curvature of the track.

References

[1] Thomas Albrecht and Jelle van Luipen. What role can a driver information system play in railway conflicts? *11th IFAC Symposium on Control in Transportation Systems*, 2006.

[2] E. Deutschl, C. Gasser, A. Niel, and J. Werschonig. Defect detection on rail surfaces by a vision based system. pages 507–511, 2004.

[3] Uwe Franke Hernn Badino and Rudolf Mester. Free space computation using stochastic occupancy grids and dynamic programming. October 2007.

[4] Z. Kim and T.E. Cohn. Pseudoreal-time activity detection for railroad grade-crossing safety. 5(4):319–324, December 2004.

[5] S. Kubota, T. Nakano, and Y. Okamoto. A global optimization algorithm for real-time on-board stereo obstacle detection systems. pages 7–12, June 2007. doi: 10.1109/IVS.2007.4290083.

[6] Kang, Dong-Joong and Jung, Mun-Ho. Road lane segmentation using dynamic programming for active safety vehicles. Pattern Recogn. Lett. Vol.24 pages 3177–3185, 2003.

[7] A. Labarile, E. Stella, N. Ancona, and A. Distante. Ballast 3d reconstruction by a matching pursuit based stereo matcher. pages 653–657, 2004.

[8] Yu-Ming Liang, Hsiao-Rong Tyan, Shyang-Lih Chang, H.-Y.M. Liao, and Sei-Wang Chen. Video stabilization for a camcorder mounted on a moving vehicle. *Vehicular Technology, IEEE Transactions on*, 53(6):1636–1648, Nov. 2004. ISSN 0018-9545. doi: 10.1109/TVT.2004.836923.

[9] G.Kecklund M. Sderstrm T. Akerstedt E.Lindberg A.Jansson E.Olsson B. Sandblad L.Kecklund, M.Ingre and P. Almqvist. The train-project: Railway safety and the train driver information environment and work situation. *2. Signalling Safety*, 2001.

[10] R. Marmo, L. Lombardi, and B. Gagliardi. Railway sign detection and classification. pages 1358–1363, Sept. 2006. doi: 10.1109/ITSC.2006.1707412.

[11] Nikolaus Mohler Milan Ruder and Faruque Ahmed. An obstacle detection system for automated trains. pages 180–185, June 2003.

[12] S. Mockel, F. Scherer, and P.F. Schuster. Multi-sensor obstacle detection on railway tracks. pages 42–46, June 2003. doi: 10.1109/IVS.2003.1212880.

[13] S. Piva, M. Zara, G. Gera, and C.S. Regazzoni. Color-based video stabilization for real-time on-board object detection on high-speed trains. pages 299–304, July 2003. doi: 10.1109/AVSS.2003.1217935.

[14] Yaser Sheikh, Yun Zhai, Khurram Shafique, and Mubarak Shah. Visual monitoring of railroad grade crossing, 2004.

[15] M. Singh, S. Singh, J. Jaiswal, and J. Hempshall. Autonomous rail track inspection using vision based system. pages 56–59, Oct. 2006. doi: 10.1109/CIHSPS.2006.313313.

[16] Zehang Sun, G. Bebis, and R. Miller. On-road vehicle detection: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(5):694–711, May 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.104.

[17] 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 872–875, 2004.

[18] Jun Xue, Jun Cheng, Li Wang, and Xiaorong Gao. Visual monitoring-based railway grade crossing surveillance system. *Image and Signal Processing, Congress on*, 2:427–431, 2008. doi: http://doi.ieeecomputersociety.org/10.1109/CISP.2008.491.