The simulation of an installation position of wayside signals using video sequences from the train cab

N. Nagamine & M. Ukai
Signalling and Transport Information Technology Division,
Railway Technical Research Institute, Japan

Abstract

This research aims to create a simulator that would generate simulated images for use when investigating where to install wayside signals along a railway track. Because the installation of wayside signals requires such considerations on as visibility from the driver's booth, the current installation measures involve that a staff member holds a simulated emergency wayside signal at a given point and the wayside signal's visibility is confirmed from a position located in a required distance. These operations require, for both the signal staff and train operation staff, the scheduling of a date for meeting on-site after a lengthy process; this is one reason why it takes considerable time before wayside signals are actually installed. To deal with this problem, we have developed a simulation method that displays a virtual driver's-eye view of wayside signals on a track by using images of signals overlaid onto the fore view out of the train. This simulation was then used to establish and adjust scheduled wayside signal positions. The simulation uses the fore view of the cabin footage (taken with a camcorder) from the driver's-eye view and the camera parameters (focal length, height) used therein to predict the three-dimensional space. This three-dimensional space is predicted by using track gauge onscreen as a limiting parameter and solving simultaneous equations. Track gauge is calculated by means of rail extraction techniques used during image processing. The simulation technique cuts back the time needed to study wayside signal positions on-site and requires no restrictions on train schedules or operation. This report describes the algorithms used and explains how to achieve verification by our test data.

Keywords: simulation, installation position, image processing, wayside signal.

WIT Transactions on The Built Environment, Vol 155, © 2014 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/CRS140091

1 Introduction

Because wayside signal installation requires visibility checks from the driver's booth, the current measures use a simulated wayside signal made out of a painted plastic bottle or other contrivance. A staff holds the dummy wayside signal at the planned position and its visibility is checked from the position located in a required distance (e.g. a point 600m away). These operations require for both the signal staff and train operation staff to schedule a date for meeting on-site after a lengthy process; this is one reason why it takes considerable time before wayside signals are actually installed.

Dealing with this problem, we have developed a simulation method that uses camcorder footage taken from the driver's booth and displays a virtual driver's-eye view of wayside signals on a track by using overlaid images. This simulation was then used to establish and adjust scheduled wayside signal positions. The simulation technique cuts back the time needed to study wayside signal positions on-site and requires no restrictions on train schedules or operation. This report describes the algorithms used and explains to verification by our test data.

2 System layout

A summary of the system is given as follows: the wayside signal position display simulator is given a range of data, including photographic images of the driver's-eye view seen through the fore of the train and positional information on the planned location of wayside signals. The simulator outputs images depicting the wayside signals overlaid on the fore view.

Broadly speaking, the simulator relies on four data processes: fore view image and kilometer matching, rail extraction from fore view image and coordinate conversion, conversion of absolute wayside signal location coordinates to corresponding fore view image coordinates, and display of overlaid wayside signals. These processes are described below.

1) Fore view image and kilometer matching
Using fore view image frames alone does not indicate at what kilometer mark the train is on the track, making a precise determination of where to install wayside signals impossible. For this reason, the images are correlated to kilometer marker points on the track.

2) Rail extraction from fore view image and coordinate conversion
Though the simulator determines wayside signal positions by comparing them with rail positions shown in the fore view image, the process of rail position extraction draws considerable load on the machine, so rail positions are not extracted every time the screen displays images. Instead, using rail position data extracted ahead of time was deemed more effective. In addition, converting rail positions from image-based coordinates to ones based on the train's orientation makes calculating wayside signal position easier. To this end, rail positions are extracted and coordinates are converted to an equivalent set of train-based coordinates.

3) Conversion of absolute wayside signal position coordinates to fore view image coordinates.
4) This routine uses the fore view image, its corresponding kilometer position on the track, the rail position, and wayside signal positions, to calculate wayside signal positions that would be within the view range of the fore image.
5) Wayside signal display
   Wayside signal images are overlaid onto the fore view image and scaled to be of proportionate size. The wayside signals' positions can be changed, added, or deleted as needed.

3 Coordinate systems

The positions of wayside signals and other objects are displayed with three parameters: absolute coordinates corresponding to kilometer designations on the track, coordinates starting at the train location, and coordinates based on the view defined onscreen. An explanation of these coordinate systems and their conversion methods is given as follows.

3.1 Coordinate systems

Coordinates based on kilometer designations on the track (absolute coordinates) are calculated based on the center of the tracks at a starting marker of 0 kilometers, with increments defined in terms of distance and height from the kilometer marker and track center. The degree of distance from the track center changes based on the corresponding train's direction of travel, with rightward direction on the x-axis defined as positive non-zero values. Relating this data to points on a map produces coordinates very close to absolute coordinates.

Train-based coordinates, meanwhile, define their point of origin at the center of the track in front of the train, at the height of the rail surface. In this system, positions are described based on the distance from the front of the train, rightward distance, and height.

Screen-based coordinates define their point of origin at the center of the image taken from the fore view of the train. Positions are described based on location on the horizontal and vertical axes with respect to the center of the image.

3.2 Coordinate conversion

The train-based coordinates and screen-based coordinates can be converted into each other using parameter sourced from the camera. Coordinates and their conversion methods are defined below.

The train-based coordinate system takes the following values: a target i and its position (x_i, y_i, z_i), the camera orientation$\left(\theta_x, \theta_y, \theta_z\right)$, angle of depression around the x-axis (θ_x), angle of roll around the y-axis (θ_y), angle of yaw around the z-axis (θ_z), and camera position (x_c, y_c, z_c). For screen-based coordinates, the values are as follows: target position (w_i, h_i), camera focal length f, and CCD size ($size_w$×$size_h$), and screen resolution (pic_w× pic_h).

3.3 (A) Conversion from screen-based coordinates to train-based coordinates (F)

Because positions in the screen-based coordinate system are represented by two-dimensional data and positions in the train-based coordinate system are represented by three-dimensional data, the conversion lacks one dimension; however, this conversion process is only implemented when converting rail positions, so the target height can be given a value of 0 when converting to a train-based coordinate system. The position in question becomes $(x_i, y_i, 0)$, making a satisfactory conversion possible.

Positions in the screen-based coordinate system (w_i, h_i) and positions along the rails in the train-based system have the following correspondence:

$$\begin{bmatrix} x_i \\ y_i \\ 0 \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} + \alpha R \left(f \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + w_i \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + h_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \tag{1}$$

R gives the camera orientation and is defined as follows:

$$\mathrm{R} = \begin{bmatrix} cos\theta_z & \text{-}sin\theta_z & 0 \\ sin\theta_z & cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta_x & \text{-}sin\theta_x \\ 0 & sin\theta_x & cos\theta_x \end{bmatrix} \begin{bmatrix} cos\theta_y & 0 & \text{-}sin\theta_y \\ 0 & 1 & 0 \\ sin\theta_y & 0 & cos\theta_y \end{bmatrix} \tag{2}$$

3.4 (B) Conversion from train-based coordinates to screen-based coordinates (F^{-1})

In the train-based coordinate system, position of a target in the three-dimensional space is given by (x_i, y_i, z_i). Camera position, orientation, and focal distance are respectively defined as $(x_c, y_c, z_c), (\theta_x, \theta_y, \theta_z)$, and f. A virtual screen of the same size as the CCD is assumed, with position values (x_i, y_i, z_i) given the corresponding variables (w_i, h_i) in the screen-based coordinate system. The underlying relationship is described by the following formula:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} + \mathrm{\alpha} R \left(f \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + w_i \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + h_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \tag{3}$$

Solving (w_i, h_i) for the unknown α yields the following:

$$\begin{bmatrix} w_i \\ f \\ h_i \end{bmatrix} = \frac{1}{\alpha} \mathrm{R}^{-1} \left(\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} - \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \right) \tag{4}$$

The formula can then be expanded for f. This allows for reciprocal conversion between the screen-based coordinate system and train-based coordinate system.

4 Fore view image matching to kilometer mark

Kilometer marks (distance designations on the track) are matched to the fore view image by way of measuring the speed of each image frame in pixels and polling image frame/pixel-level distance data. By converting this scale, image frame/actual distance (kilometer marker) data is in turn generated. This data processing routine consists of five subroutines: aerial view conversion of the fore view image, image frame/pixel-level speed measurement, image frame/pixel-level distance computation, creation of comparative data on ambient equipment positions and image frames, and final creation of image frame/kilometer marker data.

4.1 Aerial view conversion of fore view image

In order to calculate the speed of the train from the fore view image, the rate of movement of target objects in the image over a given time is computed. Accurate measurement of rate of movement is done by compensating each element so that the apparent length on the image and the actual length come out to be the same at every interval. The rail level of the track in the fore view image is therefore corrected to an aerial view which is then used to compute speed.

Aerial view conversion is achieved by specifying four points along the left and right rails (distant and proximate points) and then allowing all future conversions to be automated. In terms of camera orientation, only X-axis dip angle is assumed. Train-based coordinates are displaced, with the use of a new coordinate system defining its point of origin at the camera's focal point. In this case, the image-based coordinates and camera-based coordinates have the following relationship:

$$x_i = -\frac{Hw_i}{f\sin\theta_x + h_i\cos\theta_x} \tag{5}$$

$$y_i = -\frac{H(f\cos\theta_x - h_i\sin\theta_x)}{f\sin\theta_x + h_i\cos\theta_x} \tag{6}$$

Two sets of points at equidistant points on the left and right sides of the rails are then defined as follows: the left-top point is set asL_t : $(x_L, y_t, -H)$; the right-top point, as R_t : $(x_R, y_t, -H)$; the left-bottom point, as L_b : $(x_L, y_b, -H)$; the right-bottom point, as R_b : $(x_R, y_b, -H)$. In a similar fashion, the corresponding positions for the camera coordinates and onscreen positions are set as: L_t : (w_{Lt}, h_t) , R_t: (w_{Rt}, h_t), L_b: (w_{Lb}, h_b), R_b: (w_{Rb}, h_b) . The distance of points along the rail thus becomes

$$y_t - y_b = H\left(\frac{f\cos\theta_x - h_b\sin\theta_x}{f\sin\theta_x + h_b\cos\theta_x} - \frac{f\cos\theta_x - h_t\sin\theta_x}{f\sin\theta_x + h_t\cos\theta_x}\right) \tag{7}$$

Further, by utilizing the fact that the left and right rails are $1.067m$ apart, $cos\theta_x$, and f are obtained. Substituting this into the above equation yields $y_t - y_b$. After normalizing the four points, the ratio of x- and y-oriented lengths becomes $(y_t - y_b)/1.067$. Therefore, the four onscreen points are converted as per the following, yielding an aerial view of the image.

$$(w_{Lt}, h_t) \rightarrow \left(w_{Lb}, h_b + \frac{h_t - h_b}{1.067}(w_{Rb} - w_{Lb})\right) \tag{8}$$

$$(w_{Rt}, h_t) \rightarrow \left(w_{Rb}, h_b + \frac{h_t - h_b}{1.067}(w_{Rb} - w_{Lb})\right) \tag{9}$$

$$(w_{Lb}, h_b) \rightarrow (w_{Lb}, h_b) \tag{10}$$

$$(w_{Rb}, h_b) \rightarrow (w_{Rb}, h_b) \tag{11}$$

In Figure 1 below is shown an aerial conversion of a fore view image. From Figure 1, it can be seen that the apparent and actual distances of the rail level of the track have been corrected.

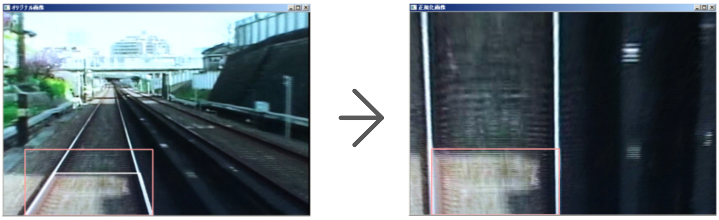

Figure 1: On-site location after aerial view processing.

4.2 Image frame/pixel-level speed calculation

After aerial conversion, the travel speed of the target object is measured with unit of pixels. This process is described below.

A scope of movement is defined for the image and speed is measured as the rate of travel per frame of video footage. Several ranges are measured in order to minimize the margin of error; after calculating several rates of travel, the ones exhibiting clear calculation errors are removed from consideration.

Pattern matching was used to keep rates of error for each scope area to the bare minimum. Of the scope areas measured, the scope exhibiting the smallest margin of error was used as a basis for computing the rate of travel. Numbers exceeding the threshold for the train's travel orientation or maximum speed were deemed errors and thus excluded. After calculating the average of the rates of travel meeting these parameters, that value was treated as the speed per single still frame. In cases where the rates of travel for each scope area were all out of the allowable threshold, the rate of travel for the previous frame would be used as the speed for the current frame.

A program based on these parameters yielded the following speed measurements for a single-station distance:

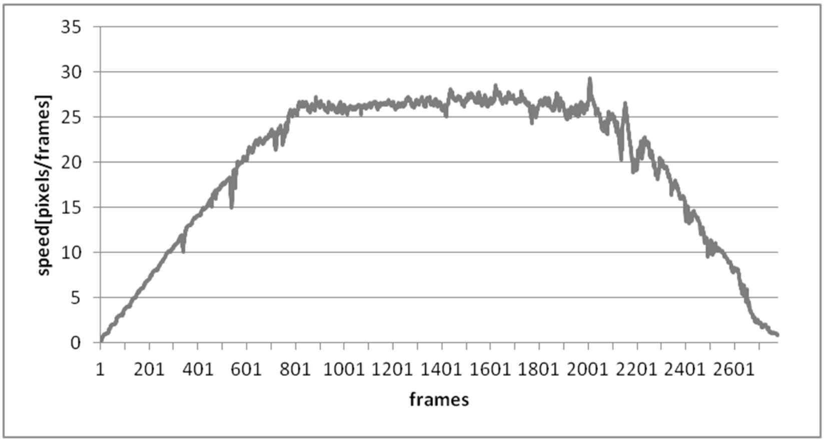

Figure 2: Speed calculation.

4.3 Image frame/pixel-level distance calculation

Integrating the speeds for each frame then yields a range of distance data expressed with unit of pixels. The distance when the footage starts is defined as 0; distance over t frames is obtained by aggregating the speeds. Computing distance via speed yields the following:

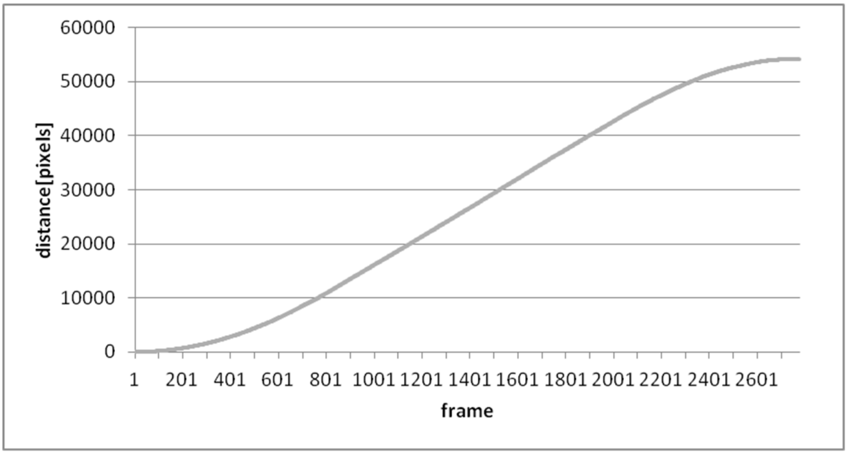

Figure 3: Frame/distance (in pixels) calculation.

4.4 Creation of reference data on equipment and frames

In order to correlate the fore view image frames with kilometer markers on the track, a table of reference data is created based on equipment with known kilometer marker locations (e.g. level crossings) and the frame number.

4.5 Creation of image frame/kilometer mark data

Using the image frame/pixel-level distance calculation data and the table on equipment locations and frames above, the image frame/pixel-level distance data is variously expanded and compressed over the intervals shown in the reference table, creating a new set of image frame/kilometer mark data. Distances are converted against the intervals by using the ratio between the distance in pixels and the actual distance displayed on the reference table.

The image frame/pixel-level distance data and ambient equipment/frame data yield a new frame/actual distance (by kilometer marker) table, as shown below:

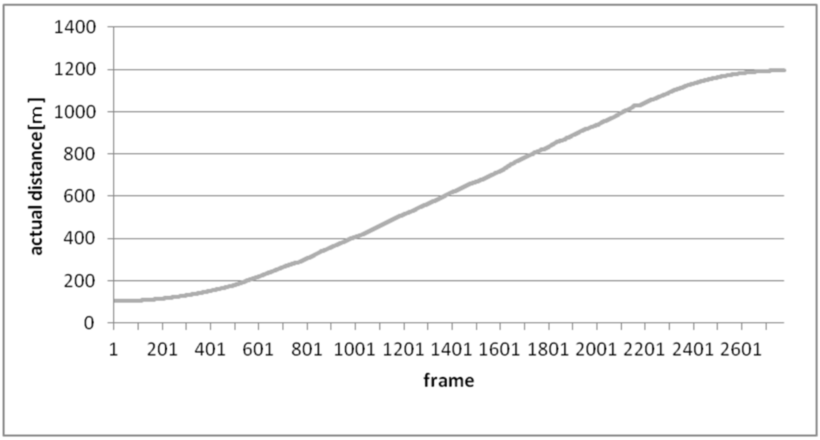

Figure 4: Image frame/kilometer mark data.

5 Rail extraction from fore view image and coordinate conversion

The rail position extracted from the image-based coordinates derived from the fore view image is next converted to a train-based coordinate system. The process of rail position extraction and coordinate conversion consists of the following steps: rail extraction, coordinate conversion, center line extraction, correction, and creation of image frame/rail position data. The rails are first extracted from the image and the center line is then obtained based on the position of the rails. This position is converted from an image-based coordinate system to the train-based coordinate system, which is based on the actual space, and the value is stored.

In order to convert the rail position from the aforementioned image-based system to a train-based one, several nodal points are defined along the extract rail position; these points are then converted to train-based coordinates and the center line is obtained.

The function for the aforementioned conversion is set as F, with the inverse function being F^{-1}. The spread between the maximum and the minimum heights of the rail is so minuscule as to be safe to ignore, so the rail height is assumed to be level.

Starting from the bottom of the screen and working up vertically, uniform nodal point values (0, 1, …) are set along the rails. Using the function F, these values are converted to positions in a train-based coordinate system. In the figure below, r corresponds to positions in the image-based coordinate system, while R corresponds to positions in the train-based system.

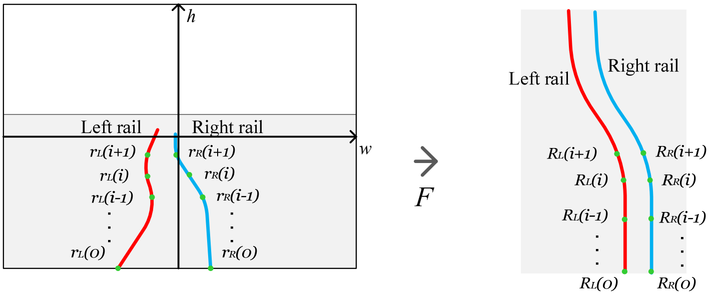

Figure 5: Creation of nodal point data for onscreen rails and actual rails.

6 Displaying wayside signals on the fore view image

Next, displaying wayside signals at the appropriate locations on the fore view image requires calculating the size of the wayside signal image to be used.

6.1 Onscreen wayside signal size calculation

The data processing routines described in Section 5 solely deal with wayside signal positions. The next step involves determining in what size the wayside signals should be displayed at the desired onscreen positions. Wayside signal image data is preloaded with the parameters of pixel size and actual wayside signal size(h_s) relative to the screen size. In addition, the baseline position for the wayside signals is based on the center of the wayside signal image.

The wayside signal position in train-based coordinates, the subject of the preceding chapters, is given here as(x_s, y_s, z_s). Using this, an average distance from the camera (x_c, y_c, z_c) to the wayside signal is given by

$$\sqrt{(x_s - x_c)^2 + (y_s - y_c)^2} \tag{12}$$

A maximum bound is defined for this distance; values exceeding that bound are deemed too far to make visual confirmation, so they are not displayed.

By using the camera data – focal distance f, CCD size $(\text{size}_w \times \ \text{size}_h)$, and image resolution $(\text{pic}_w \times \ \text{pic}_h)$, the height of the wayside signals shown on the camera CCD is given by

$$\frac{f}{\sqrt{(x_s-x_c)^2+(y_s-y_c)^2}} h_S \tag{13}$$

The onscreen wayside signals positions are thus given by

$$H = \frac{f}{\sqrt{(x_s-x_c)^2+(y_s-y_c)^2}} h_S \frac{1}{\text{size}_\text{h}} \text{pic}_h \tag{14}$$

$$W = \frac{f}{\sqrt{(x_s-x_c)^2+(y_s-y_c)^2}} h_S \frac{1}{\text{size}_w} \text{pic}_w \tag{15}$$

A minimum bound is then set. We specified that, provided the value was within the max/min threshold, displayed images would not have fewer pixels than one. In order to ultimately display the wayside signal onscreen, the original size (pic_h) is expanded or compressed as necessary to reach the above pixel amount.

6.2 Displaying wayside signals at desired positions on the fore view image

Scaling the images, as described in the prior paragraph, and replacing the pixel size at the position obtained through the routines outlined in Section 5, enables the simulator to display the wayside signals at the desired locations and in the correct proportional size. Wayside signal images are overlaid onto the fore view image irrespective of what the scene actually depicts, so there is no means of depicting wayside signals occluded by trees or other real-world obstructions.

7 Testing the validity of this simulation method

Using the actual scene from the driver's booth and the hypothetical wayside signal positions created herein, we tested the validity of this simulation technique. The image frame/kilometer mark data, image frame/track center line position data, fore view images with overlaid wayside signals, and other results of data processing were exported as files. The trial used video footage with the following characteristics: 11km of travel distance, 20 minutes of aggregate film time, 35,519 total frames, and an image size of 720 x 480 pixels. The corresponding wayside signal image had the following characteristics: an image size of 31 x 214 pixels and a corresponding real-world height of 1.5m.

To confirm the validity of the fore view image and kilometer marker matching performance, we exported speed measurements, and actual distance measurements, as shown respectively in Figures 6, and 7. These results allowed us to obtain, by means of the fore view video footage and reference data created, kilometer marker position data corresponding to each frame of the video. In this program, the accuracy of speed measurement affects the margin of error of the outputted data, so further improvements to speed measurement fidelity will yield more accurate output. In addition, the more ambient equipment is included in the reference data table, the more accurate kilometer marker data can be obtained, but discovery of ambient equipment must be done by watching the footage live and entering it into a table, so the process takes considerable time. Creating reference data more efficiently will be an issue requiring further consideration in the future.

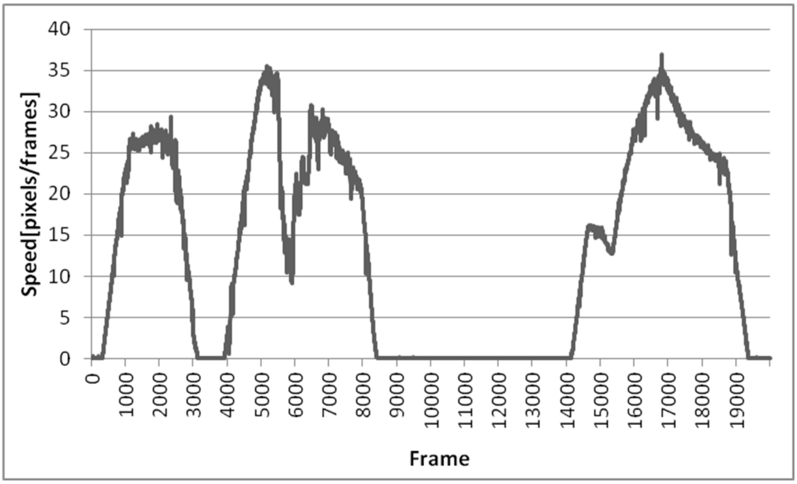

Figure 6: Results of frame/speed measurement.

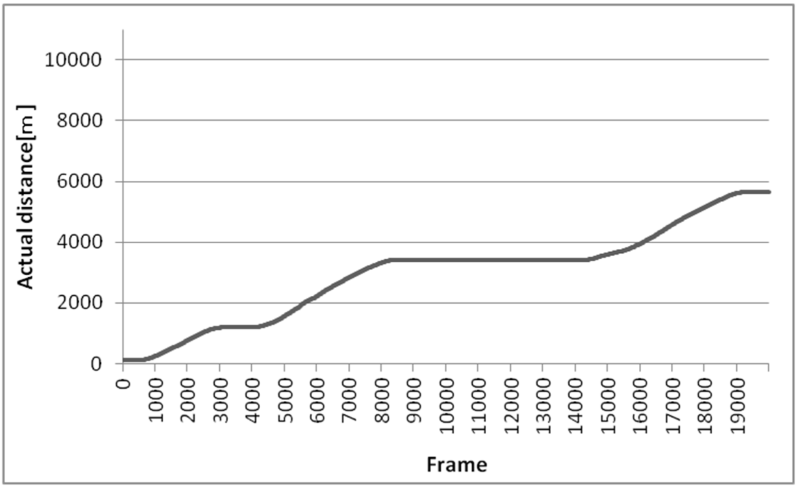

Figure 7: Results of frame/actual distance calculation.

The routines of rail extraction form the fore view image and coordinate conversion were tested by going through those respective processes and outputting the resulting nodal point coordinates along the track center line. The program (shown below) confirmed that high-fidelity results for the line had been obtained.

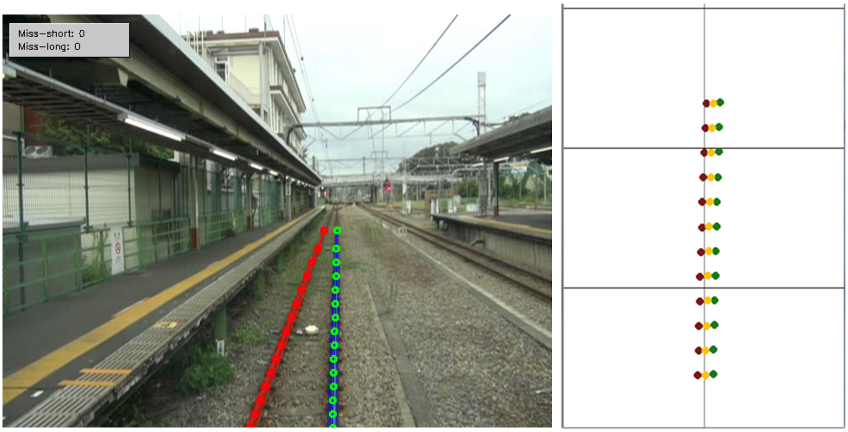

Figure 8: The program when running.

To test the wayside signals overlaid on the fore view image, we used the fore view video footage, frame/kilometer marker data, frame/track center nodal point data, wayside signal position data, wayside signal image data, and camera parameter data to export final images showing wayside signal positions at each frame, as shown below. We confirmed that the wayside signal positions at each frame were correctly converted from rail-based coordinates to corresponding screen-based coordinates.

The resulting simulator is able to run a simulation displaying potential wayside signal installation points and provides ways of determining where to ultimately install the signals on-site.

We noticed that track curving in the distance are not detected properly, causing wayside signal positions to be erroneously displayed. Because the fidelity of rail extraction at distant points in particular affects the ultimate accuracy of displayed wayside signal positions, improvements to rail detection and extraction will yield still more accurate wayside signal position candidates.

Further, because this simulator has been designed with the bare minimum control functionality and user-friendliness needed to run it, improving them is a future issue to explore.

In addition, as mentioned above, this system cannot depict situations where wayside signals are occluded by trees or other objects. Accurately depicting such situations would prove to be highly difficult given the current input parameters used with this system.

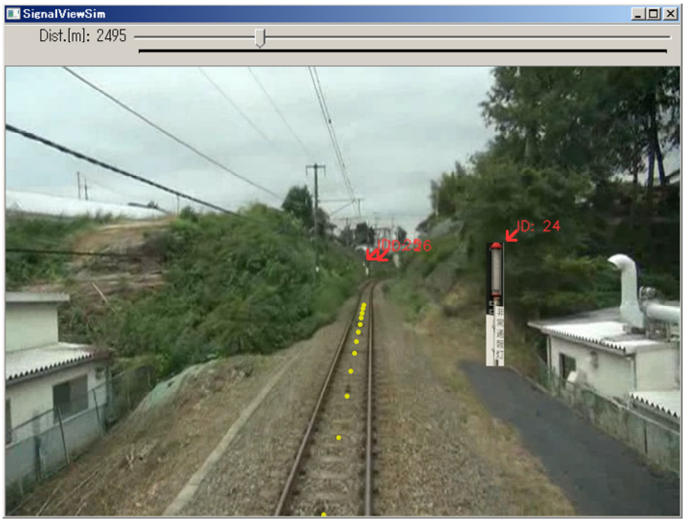

Figure 9: Displaying a wayside signal on the fore view image.

8 Conclusions

We developed this simulation in order to display a simulated driver's view of wayside signals on the railway tracks, with wayside signal images overlaid onto images depicting the view out of the fore of the cabin. The simulation was developed such that wayside signal positions could be confirmed and adjusted, thereby making the process of installing real-world wayside signals easier and more efficient. We then tested the program's performance and determined that it has more than adequate functionality for practical use.

Future issues include improving the fidelity of speed measurements obtained through image correlation, automatic detection of equipment installed on the ground, and other techniques to more efficiently create the requisite reference data tables. We should to test for performance evaluation under varying lighting conditions (poor lighting, sun light reflections, shadow overcast, etc.). Other advances could include faster rail extraction processing, higher fidelity detection of distant rails, and user interface improvements to add to or expand on the core functionality provided and to make the tool more user-friendly.

References

[1] Nagamine, N & Ukai, M. Visibility check method of obstruction warning signal, *the papers of Transportation, Electronic, and Railway, IEE Japan* 2009(1), pp. 31-35, 2009.

[2] Nagamine, N & Ukai, M. Visibility Checking Method for Railway Signals by Cab-mounted Camera, *Quarterly report of RTRI,* Vol.54 (2013), No. 2, pp. 85-90, 2013.