DOI 10.5162/ettc2020/1.1

DISTANCE MEASUREMENT FOR INTRUSIVE OBJECTS ON RAILWAY TRACKS

Jian Song, Junyuan Yang

Guangzhou Zoweetech Ltd.

Guangzhou, Guangdong, PR China

songj@zoweetech.cn

ABSTRACT

It is required to measure the distance between the moving carriage and intrusive foreign objects or the carriage ahead during shunting operations. The traditional solutions for distance measurement include laser and radar distance measurement methods, while more and more studies about the binocular vision system come up. For laser distance measurement methods, one dimensional rangefinder is feasible for the objects in a straight line ahead; however, the scanning laser rangefinder cannot distinguish whether the objects ahead are on the railway track. Limited by the measurement accuracy resulting from camera installation on board, the binocular vision system[1] is not suitable for measuring the distance of intrusive objects on tracks as well. In this paper, a monocular vision system based on special designed image pre-processing algorithm is presented, which will accurately recognize and locate the railway tracks and objects on tracks. Moreover, this system could calculate the distance between the carriage with camera and the objects detected.

1. Introduction

Shunting operation is the center of a railway yard, and is the important component of the whole railway transportation. As shunter driver could not detect carriages or intrusive foreign objects on the tracks while the train is moving backwards, a detection mechanics or system should be applied to monitor the intrusive objects and carriage on the tracks which the train is running, and to measure the distance between for safety guarantee.

The binocular method[1] for foreign object detection uses a binocular camera to measure the track area, and performs stereo matching and three-dimensional reconstruction on the images collected by the left and right cameras to detect invading foreign objects. However, due to the huge amount of calculation in the reconstruction, it is very difficult to apply binocular method. Regarding the foreign object

distinguishing algorithm, an attempt was made[2] to classify the target by extracting target features and combining with SVM classifier. Another study[3] tried to discover intrusive behaviors based on the motion tracking of moving targets. There are other achievements related as well, but all these achievements using fixed cameras are not the best choice for railway shunting.

An solution combining both railway track recognition and invading foreign objects detection is present in this paper. An onboard device with built-in camera is used to recognize the running tracks, shunting carriage and foreign objects, then to measure the distance between.

2. System Framework

The system is composed by four modules, including image acquisition, data transmission, image processing, result display and alert, among which image processing is the focus of this paper. The flow chart of the framework can be shown as figure 1:

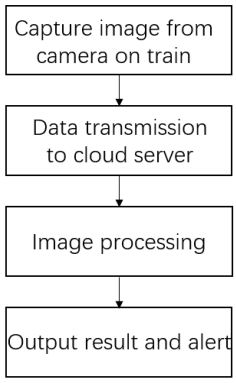


Figure 1: Flow chart of the whole system

There are two parts in image processing module, ranging and recognition. The ranging algorithm based on track detection consists of the following four steps:

1) Optimizing the edge detection algorithm to perform Canny edge detection[4].

2) Sifting to distinguish the edges of the railway track and that of the crosstie/ballast.

3) Distinguishing current running tracks.

4) Distance measurement based on perspective principle.

While the local texture features are extracted by LBP algorithm, and classified by SVM, the foreign objects are identified by their coordinates and the information of the bounding rectangular frame. Then the KCF + DSST algorithm are applied for precisely real time and multi-scale target tracking. The flow chart of image processing can be shown as figure 2:

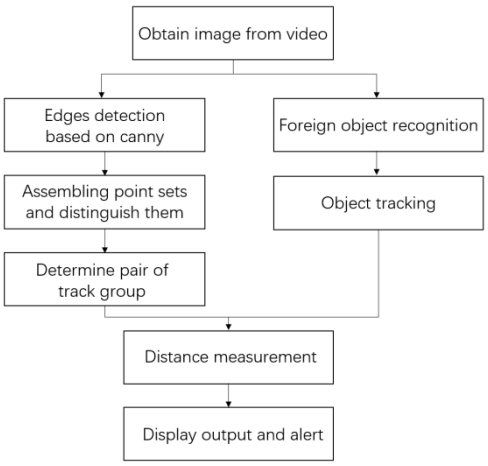


Figure 2: Flow chart of image processing

3. Railway Track Detection Based on Edges Recognition

3.1 Edge Detection Based on Canny Method

Based on Canny edge detection, an optimized edge detection algorithm with Hough line recognition is introduced in this paper to prevent unnecessary interference during image process. The edge detection operator usually distinguishes the edge of the target in the image according to the gradient mutation of the image pixel, which is mainly divided into two detection methods:

1) Based on the first derivative, the target edge detection is performed by calculating the gradient value at the corresponding pixel, such as: Prewitt operator, Roberts operator and Sobel operator.

2) Based on the second derivative, the image edge detection is performed by seeking the zero-crossing points in the second derivative, such as: Canny operator, Laplacian operator, and LOG operator.

The application of Canny edge detection algorithm[4] is divided into 5 steps:

1) Process the original image using Gaussian filter to reduce the invalid edges caused by noise.

2) Convolve the gray image to calculate the direction derivative and gradient.

3) Non-maximum suppression of the amplitude angle of the result.

4) Double threshold on the results after non-maximum suppression, and determine the pixels on the edge by the higher threshold, the lower threshold, and the connectivity of pixel.

5) Connect the edges points of the image after double threshold processing.

3.2 Meanshift Filtering

Light, shadow, crosstie etc. will cause many useless edges and noise in the image, so an appropriate filtering algorithm should be chosen for preprocessing the image. Comparing the Gaussian filtering algorithm with the Meanshift filtering algorithm, Gaussian filtering will blur the edge of the railway track and cripple the far edge detection, while Meanshift filtering can blur the texture information inside the railway tracks, but with little impact on the important track edge information.

MeanShift algorithm is essentially a general clustering algorithm. For a given number of samples, one of the samples is selected, and the sample is used as the center point to define a circular area and determine the center of mass. Then, continue to perform above-mentioned iterative process with this mass center point as the center until it finally converges.

In this paper, this feature of the mean shift algorithm is used to achieve smooth filtering of the image at the color level. This method could neutralize pixel color with similar neighbor pixel color, thereby erasing the texture details inside the railway track.

(a) Original image (b) Gray image

(c) after Gaussian Filtering (d) after Meanshift Filtering

Figure 3. Comparing original image and that after filtering

Figure 3 shows that the Meanshift filtering algorithm can eliminate the texture interference inside the object while ensuring the railway track edges not blurred.

3.3 Edge detection combined with Meanshift and Canny

With excellent characteristics of single-pixel edges and robustness, the edge algorithm based on Canny operator is effective on the edge detection for most images, so Canny operator is widely used in edge detection. However, due to the complex texture information such as light and shadow interference and crosstie/ballast edge noise in the image, there are a lot of useless edges when directly detecting the railway track edge using Canny. For an improvement, an optimized edge detection algorithm that combines Canny edge detection algorithm combined with Meanshift filtering is introduced to retain the railway track edge information to the max and to eliminate the useless edge information of background and other interfering objects. Figure 4 below compared Canny edge detection algorithm and filtering.

(a) Canny without filtering

(b) Canny with Gaussian filtering (c) Canny with Meanshift filtering

Figure 4: Results of different Canny filtering method

3.4 Point Assembling Algorithm

Since the result from Canny edge detection is a binary image containing edge information, the scattered points cannot reflect the continuity of the track edges, and cannot determine whether a edge point belong to the track or belong to the interfering. An algorithm for point clustering and improves the search method based on the traditional 8-domain search method is proposed in following.

Number 8 field directions in order from 0 to 7 as figure 5:

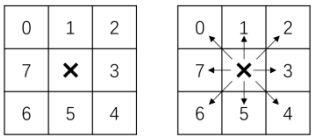


Figure 5: Non-zero center point and next non-zero point

As shown in figure 6, the connection status can be classified into two types: stable connection and virtual connection. The stable connection is in the positive direction of the valued point, that is, a non-zero point located on the 1,3,5,7 direction. The virtual connection is the valued point in the oblique side direction, that is, on the 0,2,4,6 direction lies a non-zero point.

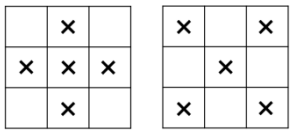


Figure 6: Stable connection and virtual connection

The improved 8-domain search method can reduce the number of times to find the next value point at a certain extent, thereby improving search efficiency, and speeding up the performance of the algorithm. The operation process is shown as follows:

1) Search non-zero points in the image from left to right and from bottom to top. The first point found will be used as the initial point, and the first current point. A new point set class is set up to store the attributes such as the position coordinates of the continuous non-zero points set growing from this initial point.

2) Look for the non-zero points in the stable connection direction of the current point. Store the new non-zero point to the list of non-zero points contained by the point set class and store the current point if there is one, then set the current point value as 0.

3) If there is none, then search the non-zero point in the virtual connection direction. Store the point in the list of non-zero points and set the current point value to 0 if there is.

4) If no non-zero points can be found in the stable connection direction and the virtual connection direction, the current value point is determined as the endpoint.

5) Select the next point from the list of non-zero points as the next current non-zero point.

6) Repeat above steps until all valued point sets in the image are generated.

In a experiment, a total of 17-point sets were found without the optimization process. The running time of the point assembling algorithm is 0.25531 seconds, the total time for processing the single image is 0.41285 seconds.

After optimization by algorithm above, a total of 18-point sets were found. And the time consumed for the point assembling algorithm was 0.16253 seconds, the total time for processing the single image is 0.33530 seconds. The speed of point assembling algorithm is increased to 36.34% while the speed of processing a single image is increased to 18.78%.

The output image of the improved point assembling algorithm can be shown as figure 7. The white line is the searched non-zero point set, and the red points indicate the endpoints.

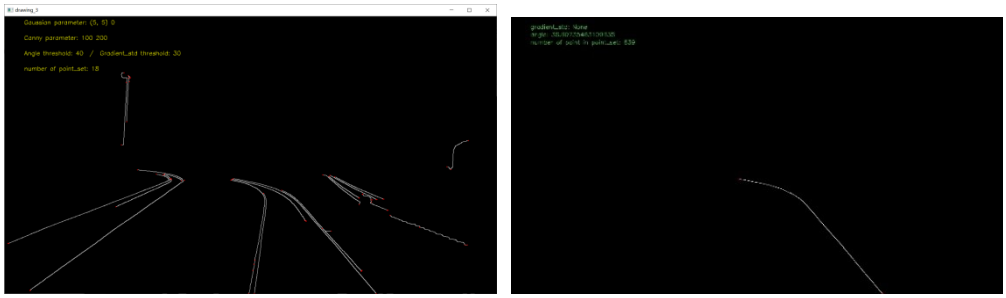


Figure 7: Left is the Canny edge detection result, right is one of the recognized point set

3.5 Delineating Region of Interest with Probabilistic Hough Transform

In the field of computer vision, it is common to complete straight line detection by Hough transform algorithm[5], which maps the plane of the rectangular coordinate system to the curves of the polar coordinate system and maps the straight line to cross over point of the polar coordinate system. In this paper, the improved Probabilistic Hough Transform method, that is Progressive Probabilistic Hough Transform, prompt by Xu[6] is used, which alleviates the shortcomings of excessive storage occupation in standard Hough Transform.

In order to solve the problem caused by the complicated background, a ROI selection method combined with probabilistic Hough Transform is adopted in this paper to constrain the initial point of the point set. Thereby speeding the point assembling.

Generally, the process of probabilistic Hough Transform can be illustrated as bellow:

1) Input the image after edge detection by optimized Canny Algorithm.

2) Randomly sample the edge points in above image.

3) Transform the selected points by Hough and vote in the accumulator space formed by the distance ρ and the angle θ ($0 \sim \pi$)

4) When the threshold is reached, it is considered that a straight line exists, and move along the straight line direction to find the two endpoints of the straight line.

Select appropriate Hough Transform parameters to obtain a series of straight lines, which are classified as "right-inclined straight line" and "left-inclined straight line" by their slope. These straight

lines in different situations require different methods of constructing ROI.

For a right-inclined straight line, the coordinates of the upper left endpoint [x1-5, y1-5], [x1 + 5, y1-5] form the two endpoints of the parallelogram; the coordinates of the lower right endpoint [x2-5, y2 + 5] , [X2 + 5, y2 + 5] forms the other two endpoints of the parallelogram.

For a left-inclined straight line, the coordinates of the lower left endpoint [x2-5, y2-5], [x2 + 5, y2-5] respectively form the two endpoints of the parallelogram. The coordinates of the lower right endpoint [x1-5, y1 + 5], [x1 + 5, y1 + 5] form the other two endpoints of the parallelogram.

ROI constructing method can be shown as figure 8:

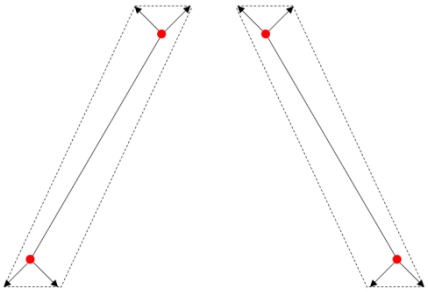


Figure 8: ROI constructed by right-inclined straight line and left-inclined straight line

The parallelograms after expanding all the straight line segments in the probabilistic Hough straight line detection result are represented in an image, named ROI_Mask, as shown in the figure 9:

Figure 9: Region of interest for point assembling algorithm

The initial point of the point assembling algorithm should have the non-zero pixel at the position in the Canny edge detection map, and the non-zero pixel at the corresponding position in the ROI_Mask image. This approach eliminates the set of interference points generated by the edges of the crosstie/ballast, background, etc.

Without ROI delineation in the experiment, the point assembling algorithm needs to search 982 times, and 31 point sets are found. The point assembling had taken 0.24538 seconds, while a single image process costed 0.42791 seconds, and the result of point clustering is shown in the figure 10:

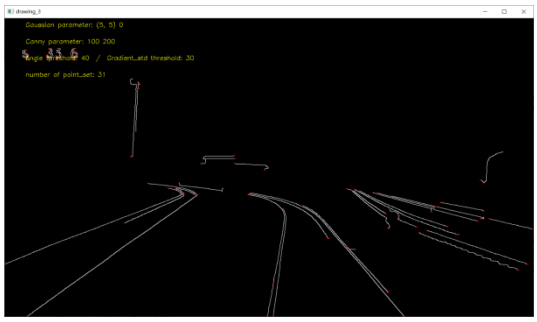


Figure 10: Point assembling result without improvement

If ROI delineation algorithm is applied to limit starting point selection, there are 84 searches in the point assembling, with a total 18 point sets found. The point assembling takes 0.15757 seconds, and the single image process takes 0.33620 seconds. The speed of point clustering is increased by 35.74%, and the speed of single image process is increased by 21.43%.

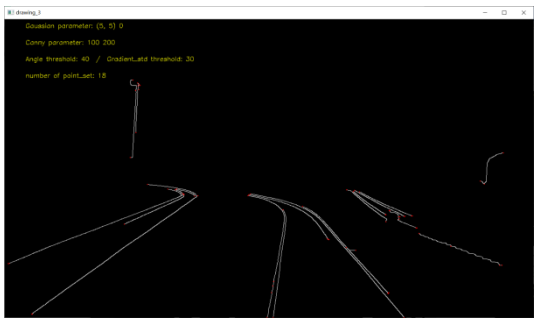


Figure 11: Point assembling result with ROI optimizing

4. Track Pair Determination Algorithm

The assembled point set is obtained after above steps, and points in the set are continuous. However, we still cannot distinguish point sets of the Railway Track from the ballast or other interference term.

In fact, the edge point set attribute can be used to preliminarily separate the track edge from most of the background interference items such as crosstie/ballast edge. In addition, the edge point sets belonging to the same track are merged and smoothed, each track has its unique corresponding edge point set. A score table could be generated by judging whether each two edges constitute a track group, to indicate which two tracks constitute a track group.

4.1 Determine Railway Track Point Set

After a series of point sets obtained from point assembling, we need to distinguish railway track point set from ballast point set by checking the attributes of each point set. Four attributes listed below, derived from the continuity of the edge point set, gradient similarity and other characteristics, are set as the criteria for distinguishing the railway track point set from the interference items.

1) The number of non-zero points in the point set.

2) The height difference between the highest point and the lowest point of the point set.

3) The angle value of line formed by the starting point and the ending point.

4) The standard deviation of the gradient of all points in the point set

Various threshold combinations were tried in the experiment, the threshold of the number of points in the point set is 100, the angle threshold is 40°, and the gradient standard deviation threshold is 20. The flow chart of sifting processing is as figure 12:

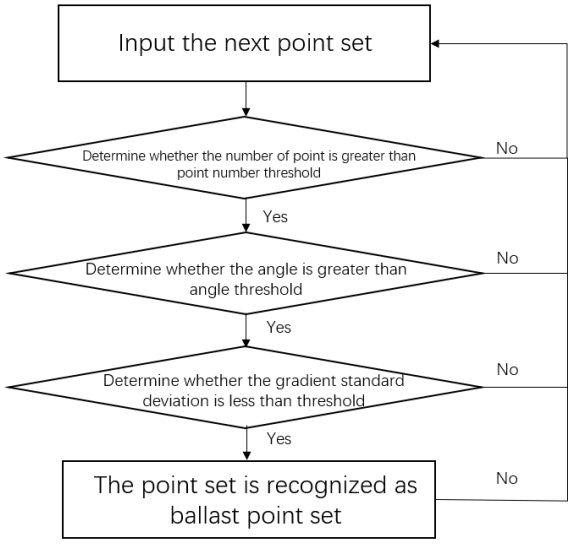


Figure 12: Flow chart for sifting processing

In the case where the edge point set is not sifting, in addition to the needed railway track point set, there are point set of interference objects such as background, as shown in figure 13:

Figure 13: Point set without sifting

After using the sifting algorithm, in figure 14, the railway track point set is retained, and most of the crosstie/ballast edges and background edges are removed:

Figure 14: Point set with railway track threshold sifting

4.2 Determine Crosstie Edge Point Sets

As a pair of tracks lies on a set of , it is possible to identify a pair of tracks by the position of the crossties. Having set some thresholds, 20°for angle, 20 for height, 5 for aspect ratio, 20 for gradient standard deviation, the crosstie edge determining process is shown as figure 15:

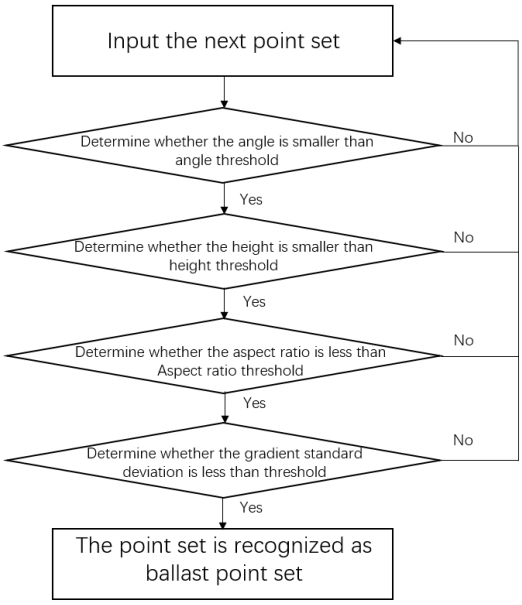


Figure 15: Flow chart for determining crosstie edge point set

The recognition result is shown as figure 16:

Figure 16: crosstie edge point set recognizing result

Corresponding to the original picture, in figure 17, it can be found that there are some background interferences identified as crosstie, but the basis for distinguishing the track pair is the number of crossties in the middle of the two tracks. So a few crossties misrecognition will not affect the result of the track pair recognition.

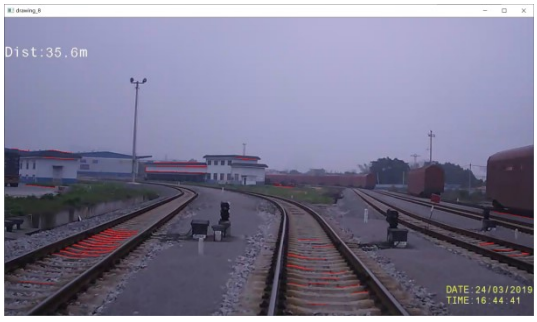


Figure 17: crosstie edge point set located in original image

4.3 Merging and fitting of Point Sets of the Same Railway Track

Observing the result of railway track sifting recognition, it can be seen that the same rail has two left and right edge lines due to the influence of light, and when it is disturbed, one track may be represented as more number of point set. Therefore, an algorithm needs to be designed to merge all edge point sets of the same track and fit a smooth curve to represent the track position.

There are the following steps to merge the point sets of the same track:

1) Sort railway track point set according to the intercept at the bottom of the image.

2) Determine that the set of points with a difference in intercept of less than 20 pixels is the edge of the same track, and merge them by merging the points at the same horizontal height (same y). The x coordinate of merged point is formed according to weighted average using the gradient size.

3) Perform multiple function fitting and smoothing on the points after weighted average.

The result can be shown as Figure 18.

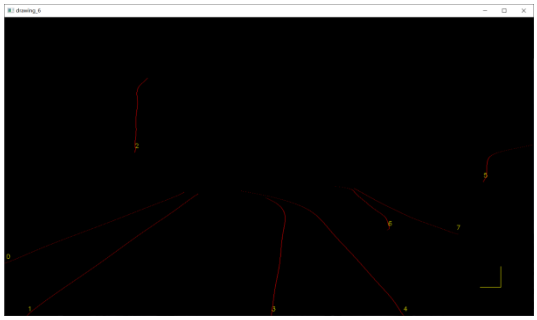


Figure 18: Point set after merged by railway track

Corresponding to the position in the original image (as shown in the Figure 19), it is found that the set of fitting points is roughly located on the track centerline, which can be used as the basis for the next step, distance measurement based on the railway track spacing and perspective principle.

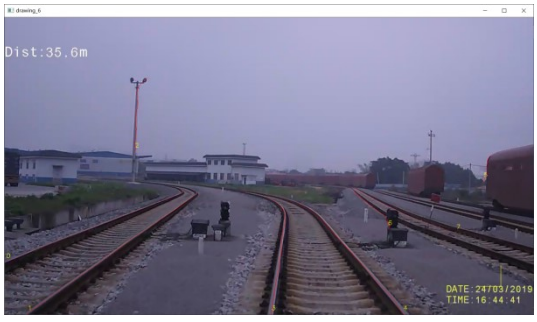


Figure 19: Merging and fitting result shown on original image

4.4 Track Pair Distinguish with Crosstie Edge Information

After obtaining the track coordinates, it is necessary to further determine which two tracks belong to one track pair. It can be found by observing the Canny result that there are crosstie edge in the middle of track pairs. So this paper proposes a track pair determination method taking crosstie edge information into consideration.

First calculate the position of the gravity of each crosstie point set; then randomly select two **point sets i** and **point set j** with similar height as the point set identified as the railway track. Finally, determine the height range for the two point sets, that is, the minimum value of the highest height of the two point sets is selected as the upper limit of the common height range, and the maximum value of the minimum height of the two point sets is selected as the lower limit of the common height, as shown in the figure 20.

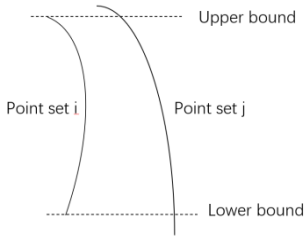


Figure 20: Sketch map for two railway point set and their common height range

Take two points at the same height within the common height range, and calculate their midpoint. Calculate the abscissa difference **x_delta** from the midpoint to the gravity of the crosstie point set with the same ordinate value. That difference, **x_delta**, can be used as the scoring criteria for judging whether **point sets i** and **point set j** belongs to the same track group.

Table 1 Score contribution by different x_delta

Distance	Score
x_delta < 10	50
10 ⩽ x_delta < 20	40
20 ⩽ x_delta < 30	30
30 ⩽ x_delta < 40	20
40 ⩽ x_delta < 70	10
70 ⩽ x_delta < 100	5

The more the crosstie gravity between **point set i** and **point set j**, the more the crosstie center of gravity will contribute to "score of the same track group" for **point set i** and **point set j**. The track group is determined by the final formed score table. If the scores of the track edge **point set i** and **point set j** are the highest in the row of **point set i**, as well as in column **point set j**, it is determined that the track edge **point set i** and **point set j** are two rails of the same set of tracks .

As shown in Figure 21, there are a large number of crossties between point set 0 and point set 1, it

can be determined that these two point sets form the same track group, and the (0,1) position and (1,0) position in the constructed score table has a score of 735; there are a large number of crossties between point set 3 and point set 4, it can be determined that these two point sets form the same track group, and score in the (3, 4) position and (4, 3) in the constructed score table is 260.

Table 2 Score table constructed for judging track group

Point set i / Point set j	0	1	2	3	4	5	6	7
0	0	735	0	0	0	0	0	0
1	735	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	260	0	0	0
4	0	0	0	260	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

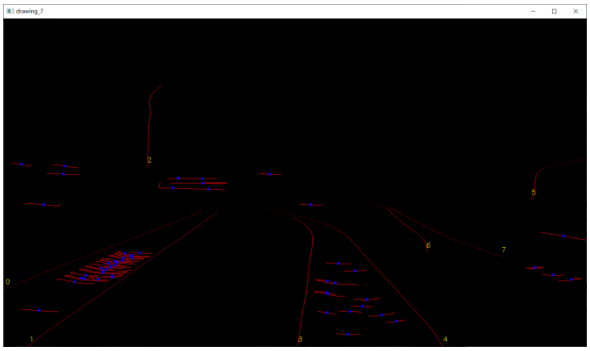


Figure 21: Red lines are the recognized railway track edges and crosstie edges, blue points indicate the position of gravity of crosstie; according to their positional relationship, the track group can be determined

5 Foreign Object Identification and Tracking

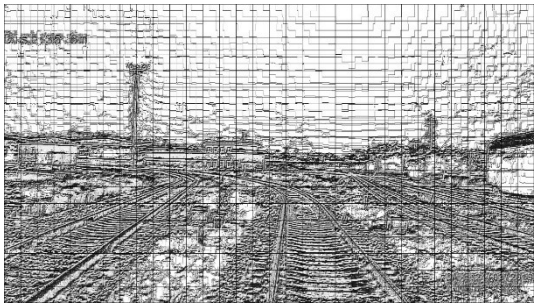
5.1 Feature Extraction Based on LBP

LBP, Local Binary Pattern, is an operator for describing local texture features in images and is widely used in face recognition, expression recognition and other fields[7]. In this paper, LBP is used for extracting the feature of railway track, crosstie/ballast and unknown objects (background).

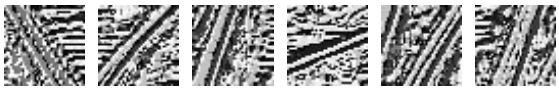
The circular LBP operator can freely adjust the sampling radius and specify the number of sampling points to be used. For example, $LBP_{8,1}$ refers to sampling 8 pixels on a circle with a sampling radius of 1. If the gray value of the sampled pixel is higher than the center gray value, it is marked as 1, otherwise marked as 0, and finally a set of binary sequences would be outputted. Convert this binary sequence to decimal to get the LBP value of the center pixel.

Because the LBP feature calculated by the circular LBP operator used in this paper has rotation

invariance and gray invariance, it can be well adapted to the misjudgment of foreign objects that may be caused by light and shadow interference. After processing the image using the LBP operator, an LBP "code" can be obtained at each pixel, and the extracted LBP feature is still an image. The image is divided into 40 * 40 small blocks, which is convenient for subsequent SVM classification processing. The LBP feature map is shown in the figure 22.



(a) LBP feature for the whole image



(b) LBP feature for railway track block

(c) LBP feature for crosstie block

(d) LBP feature for unknown objects (background)

Figure 22: LBP feature map for the whole image, railway tracks, crosstie and unknown objects

5.2 Foreign Object Recognition by SVM Classifier

SVM[8] is a classifier that can achieve better based on fewer sample amount. Its basic model is defined as a linear classifier with the largest interval on the feature space. Its learning strategy is to maximize the interval, which can be converted into a convex quadratic programming problem[9]. The SVM algorithm was originally designed as a binary classifier. In this paper, the one-versus-rest SVM is applied, and a multi-classifier is constructed by combining multiple classifiers.

In this paper, the extracted LBP feature parameters are used as a classification basis, by which all the small blocks are manually divided into three categories: rails, crosstie/ballast, and unknowns (background). During AI training, one kind of sample is classified into one class as a positive set, and the remaining as a negative set. Three kinds of samples produce three SVM classifiers. During the test,

the test vectors are tested with three trained SVM classifiers respectively, and three scores are obtained, and the largest one of the three scores is taken as the classification result.

The image is divided into multiple blocks of 40×40 pixels, and each block is classified into a training set and a test set of three kinds of samples. There are 1458 blocks in the track test set and 516 blocks in the test set; 1549 blocks in the crosstie/ballast training set and 605 blocks in the test set; 2998 blocks in the background training set and 2140 blocks in the test set.

The model file **clf.model** obtained from the training set was used to test the recognition accuracy. The accuracy of the track recognition was 0.87, the accuracy of the crosstie/ballast recognition was 0.89, and the accuracy of the background recognition was 0.97.

The result of recognizing the whole image is shown in the Figure 23 below, where label 1 indicates recognition as a track, label 2 indicates recognition as a crosstie/ballast, and label 3 indicates recognition as an unknown object (background).

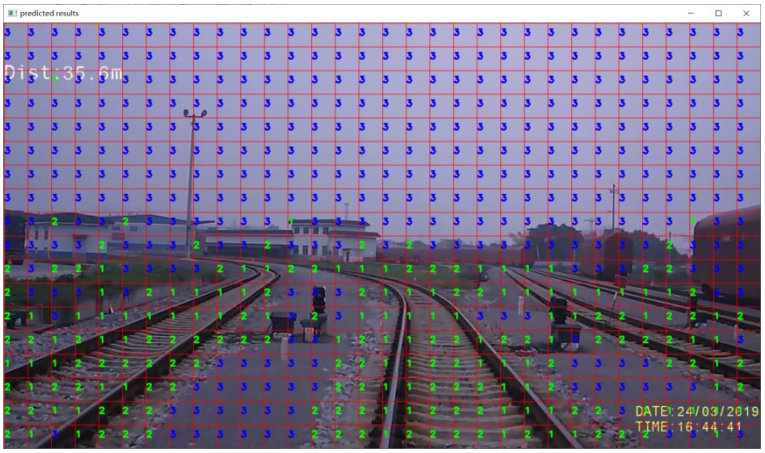


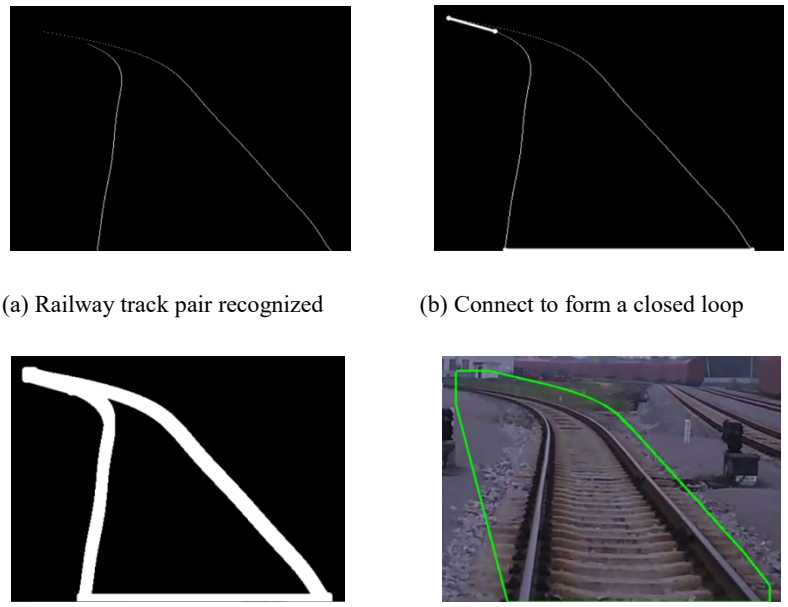
Figure 23: Recognizing the whole image

However, classifying all the small blocks of the full image will cause excessive time consumption. Usually, the detection of a whole image will take about 3 to 4 seconds, which cuts down real-time performance of the algorithm. This paper proposes a warning area division method based on railway track pairing.

First, extract a pair of tracks, as in Figure24 (a), then connect two highest points of the two tracks of the track group, and two lowest points to form a closed loop, as shown in Figure 24 (b).

Second, perform a dilation algorithm for the closed loop in order to expand the single pixel identified by the Canny edge detection to multi-pixel width, as shown in Figure 24 (c). Then look for the convex hull of the expanded graphics, and observe that the convex hull contains the track and ballast area, which will be used as the early warning area, the corresponding area in original image is shown in Figure 24 (d) .

The SVM only processes small blocks in the early warning area. If an unknown object (background) is detected, an alarm would be issued and the coordinate position of the unknown object (background) area is sent to the tracking module for tracking foreign objects.



(a) Railway track pair recognized (b) Connect to form a closed loop

(c) Processing dilation (d) Early warning area in original image

Figure 24: ROI defining process

5.3 Foreign Object Tracking Algorithm Based on KCF-DSST

Since direct foreign object detection algorithm for each frame of image will cause heavy overhead, a foreign object tracking algorithm based on correlation filtering is added to the foreign object detection algorithm. The foreign object intrusion algorithm is processed every 1 second / 15 frames, and the rest frame is using the tracking algorithm that keeps track of foreign objects. This method can ensure the accuracy of the algorithm and the timely detection of foreign body intrusion, and ensure the real-time performance of the algorithm as well.

The tracking module uses the KCF, kernelized correlation filters[10], to track the results of foreign object detection, which better target multi-scale transformation using discriminative scale space tracking method proposed by Martin Danelljan and Fahad Shahbaz Khan in 2016[11].

The algorithm has designed two consistent correlation filters to achieve target tracking and scale transformation, defined as translation filter and scale filter respectively. Translation filter is responsible for positioning the current frame target, and scale filter is used to estimate the current frame target scale. Two filters are relatively independent, so we can select different feature types and feature calculation methods to train and test.

In practical applications, the location and scale information of rectangular frame tracking are selected based on foreign object recognition, and the target position is found through the translation

filter calculation response in next frame. After the scale filter calculation response, the target's next frame scale is found. The result of the filter is used as the updated target position and target scale, and then the updated tracking rectangle is used for training and optimizing the translation filter and scale filter, and the above steps are looped to achieve target tracking. The tracking result can be shown as Figure 25.

The processing speed of the tracking algorithm is about 60fps

Figure 25: Tracking Using KCF-DSST

6 Ranging Algorithm Based on Track Group Detection

The distance measurement based on the perspective principle mainly depends on the relationship between the target distribution in the actual scene and the corresponding target position in the image. The width of the rail in the actual scene is certain, and the width of the rail in the image decreases with the distance increasing due to the perspective principle. The relationship between those two can be used to calculate the actual distance from the camera to any position on the rail.

As shown in Figure 26, in the schematic vertical view of the actual scene, the point O is the camera position on the train, and the two red straight lines represent the railway track which the train is running at; E, F are the two points on the railway track that initially entered the camera screen and actual distance from EF to camera is l_0; the two points A and B are the positions of the left and right vertex that can be seen at the bottom of the image in the actual scene; the blue object in Figure 26 represents the foreign objects appearing on the rail, and the two points G and H are the two points of the rail where the foreign object appears, the distance from the invading foreign object to the camera is l_1; the two points C and D represent the corresponding positions in the actual scene of the most left and right points that can be observed on the horizontal line of G' and H' in the Figure 27. The distance from the connection lines of C and D to the camera is also l_1; the width of the rail adopts the international standard gauge, that is 1435mm.

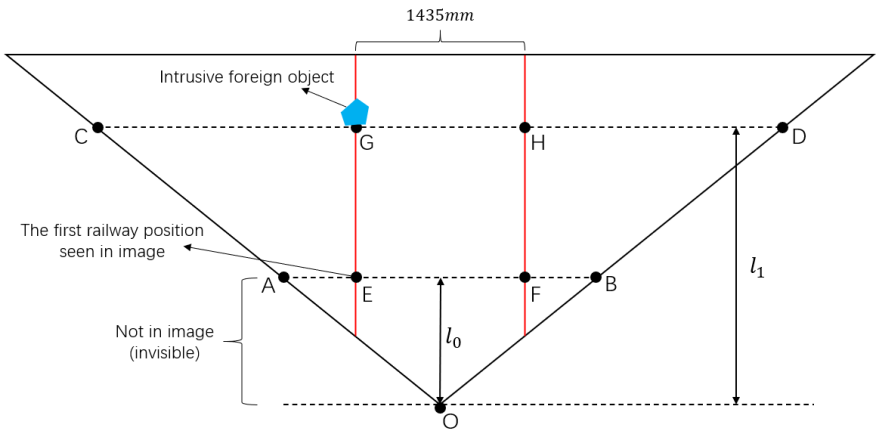


Figure 26: A schematic vertical view of the actual scene, showing the actual positional relationship between the camera, the railway track and intrusive foreign objects

In the image collected by the camera, due to the perspective principle, the width of the railway track (the red curve) continues to shrink in the picture. As shown in Figure 27, suppose the width of the image is W and the image size is 1280×720, so $W = 1280$. The E', F' points in the image represent the intersection of the railway track and the bottom of the image. The A', B' points are the left vertex and the right vertex at the bottom of the image; the intrusive foreign object is found at the G' point in the image, and the H' point is the same height of G' and on another railway track; the position of the two points C', D' in the image is the intersection of the horizontal line G'H' and the left and right borders of the image.

The abscissa difference between E' and F' is set as w_0, and the abscissa difference between G' and H' is set as w_1.

Observing the schematic vertical view of the actual scene, since the triangle OAB and the triangle OCD form a pair of similar triangle, the ratio of the CD side to the AB side is equal to the ratio of the distance l_1 from CD to point O and the distance l_0 from AB to point O, given by Equation 1:

$$\frac{l_{CD}}{l_{AB}} = \frac{l_1}{l_0} \tag{1}$$

The actual length of l_0 is roughly 7.9m, and the distance between the intrusive object and the camera can be obtained by calculating the ratio of l_{CD} with l_{AB}.

The CD line segment in the actual scene is shortened to the C'D' line segment in the image, and the GH line segment is shortened to the G'H' line segment. Considering that the actual scene is shot by the camera, the scaling ratio of objects from the same distance is the same, the equation can be deduced as:

$$\frac{l_{CD}}{W} = \frac{1435mm}{w_1} \tag{2}$$

Similarly, the EF line segment is shortened to the E'F' line segment, and the AB line segment is shortened to the A'B' line segment, and the equation can be written as:

$$\frac{l_{AB}}{W} = \frac{1435mm}{w_0} \tag{3}$$

Combining above three equations, the expression of the distance l_1 of the intruder from the camera can be derived as

$$l_1 = \frac{w_0}{w_1} \times l_0 = \frac{X_{F'} - X_{E'}}{X_{H'} - X_{G'}} \times l_0 \tag{4}$$

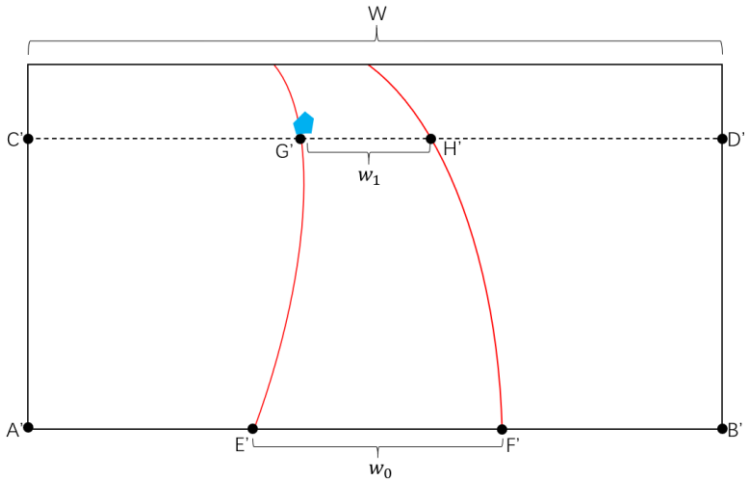


Figure 27: Schematic diagram of camera shooting image

7 Conclusion

In this paper, we designed a system to detect and track the intrusive foreign objects on railway tracks, and a distance measuring method based on railway track group recognition. In order to promote the effectiveness and real-time ability, we optimized the Canny edge detection algorithm, point set assembling algorithm and SVM classification algorithm. Furthermore, we proposed the track group classification algorithm and up to now no such algorithm is in use.

There are some problems to be solved in future:

1) Due to the complexity of the actual track scene, there will often be interference and occlusion, which leads to some reduce of the accuracy rate.

2) The limitation of the detection algorithm makes it hard to distinguish the complex rail sections such as turnouts.

3) Due to the limitation of network bandwidth, it is difficult to transmit high-definition video images for processing, which has a negative impact on long-distance foreign object detection and rail detection

The next research direction is to improve the accuracy of foreign object recognition and optimize the algorithm, improve the overall efficiency of the algorithm, and speed the algorithm to better meet the actual real-time requirements.

Reference

[1] Ohta M. Level crossings obstacle detection system using stereo cameras. Quarterly Report of RTRI (Railway Technical Research Institute), 46(2): 110-117, 2005.

[2] L. Tong, L.-Q. Zhu, Z.-J. Yu, et al. Railway obstacle detection using onboard forward-viewing camera. J Transp Syst Eng Inf Technol, 12(04), pp.79-83, 2012.

[3] Dong, H., Ge, D., Qin, Y. and Jia, L., Research on railway invasion detection technology based on intelligent video analysis. Zhongguo Tiedao Kexue, 31(2), pp.121-125, 2010.

[4] Canny, J., A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence, (6), pp.679-698, 1986.

[5] Shapiro, S.D., Feature space transforms for curve detection. Pattern Recognition, 10(3), pp.129-143, 1978.

[6] Xu, L., Oja, E. and Kultanen, P., A new curve detection method: randomized Hough transform (RHT). Pattern recognition letters, 11(5), pp.331-338, 1990.

[7] Ahonen, T., Hadid, A. and Pietikainen, M. Face description with local binary patterns: Application to face recognition. IEEE transactions on pattern analysis and machine intelligence, 28(12), pp.2037-2041, 2006.

[8] Cristianini, N. and Shawe-Taylor, J., An introduction to support vector machines and other kernel-based learning methods. Cambridge university press, 2000.

[9] Zhang, Hongshuai, Zhiyi Qu, Liping Yuan, and Gang Li. "A face recognition method based on LBP

feature for CNN." In 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pp. 544-547. IEEE, 2017.

[10] Henriques, J.F., Caseiro, R., Martins, P. and Batista, J., High-speed tracking with kernelized correlation filters. IEEE transactions on pattern analysis and machine intelligence, 37(3), pp.583-596, 2014.

[11] Danelljan, M., Häger, G., Khan, F.S. and Felsberg, M., Discriminative scale space tracking. IEEE transactions on pattern analysis and machine intelligence, 39(8), pp.1561-1575, 2016.