PLANIMETRIC RAIL POSITIONING USING UAV PHOTOGRAMMETRY: TOWARDS AUTOMATED AND SAFE RAILWAY INFRASTRUCTURE MONITORING

S. Cuypers*, H. De Winter, M. Bassier, M. Vergauwen

Department of Civil Engineering, Geomatics Section, KU Leuven, 9000 Ghent, Belgium
suzanna.cuypers@kuleuven.be

KEY WORDS: UAV, Photogrammetry, edge detection, aerial survey, rail monitoring, rail survey.

ABSTRACT:

The maintenance of railway infrastructure requires detailed inspection of track assets including the rails, sleepers, fasteners, and tie plates. Current methods using total stations and measurement trains are costly and the subsequent data processing is often manual. A potent alternative is the use of Unmanned Aerial Vehicles (UAV) to investigate track asset deviations. However, the potential degree of automation and the overall accuracy is still very much the subject of ongoing research. In this work, a potential pipeline is investigated for the planimetric inspection of rails using UAV photogrammetry. Specifically, state-of-the-art line detectors such as Holistically-Nested Edge Detection, DexiNed, Segment Anything Model, and Mobile Line Segment Detection are combined with logical filtering to assess the localisation and gauge of the rails. The experiments indicate that the accuracy and detection rate are promising. Overall, the proposed method is a promising step towards affordable and safe railway infrastructure inspection.

1. INTRODUCTION

The inspection of railway infrastructure includes the inspection, measurement, and localisation of various railway components including the rails, sleepers, fasteners, and tie plates. The localisation of the rails and the gauge needs to be measured on a regular basis. A metric deviation analysis is performed between the measured positions of the rails and the as-designed location provided as a set of 3D CAD plans. Specifically for rails, there are strict tolerances to comply with maintenance and safety regulations. Conventional ground measurement techniques with total stations, dedicated measurement trains, and Lidar techniques are time and cost intensive. Furthermore, they require rail access which interrupts train traffic and puts surveyors in potentially dangerous situations. Currently, both the data processing and interpretation are performed mostly manually, which is slow and costly, even for the measurement train in some cases. For instance, while rail track geometry measurement tools such as trolleys or measurement trains can accurately measure the positioning and geometry, every single track has to be traversed individually which significantly lowers its efficiency near stations where many tracks converge.

A promising alternative is to use Unmanned Aerial Vehicle (UAV) footage to perform similar analyses. Besides automating the process without interrupting train traffic, UAV techniques are low-cost, less labor-intensive, and safer than conventional ground measurement techniques with total stations, measurement trains, and rail trolleys. With modern technology, ultra high resolution can be achieved as shown in Figure 1, which allows for the evaluation of rail positioning with computer vision techniques. However, there are significant concerns regarding the accuracy and detection rate of UAV-based methods. The goal of this work is, therefore, to investigate the viability of automated UAV-based rail inspection. Specifically, we present an unsupervised pipeline that detects the edges of rails and compares them to their digital twin counterpart in the utility companies' database.

* Corresponding author

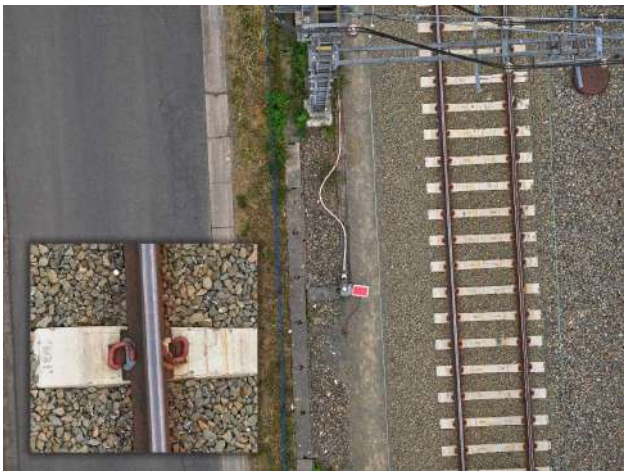

Figure 1. Drone image of 100M pixels at an elevation of 25.8 m with a detail showing the ultra high resolution: 1.17 mm GSD

In summary, our contributions are:

1. Development of a workflow for planimetric monitoring of rails using a photogrammetric UAV method.
2. Visualisation of the relative localisation and gauge of rails.
3. Adoption of state-of-the-art edge and line segment detection methods such as HED, DexiNed, Mobile Line Segment Detection (M-LSD) and Segment Anything Model (SAM) for rail boundary detection.
4. Proposal of an evaluation and filtering method for detected lines.

The remainder of this work is structured as follows. The related work is presented in Section 2. In Section 3 and 4, the data and methodology are described. The experimental results are described in Section 5. Finally, we conclude with a discussion (Section 6) and conclusion (Section 7).

2. RELATED WORK

The main concerns in railway monitoring are defect detection, track condition maintenance and deviation monitoring from the as-designed rails. Specifically for rail geometry, techniques using points clouds and imagery have been explored in the literature to improve inspection and localisation. Ariyachandra and Brilakis (2020) use airborne Lidar point clouds to generate a digital twin of railway track beds and the surrounding superstructure. The obtained RMSE of their geometric digital twin of the rail masts is 2.72 cm, which does not meet the required accuracy to determine absolute positioning of rails. Jwa and Sonh (2015) use point cloud data collected by a Lidar mobile mapping system. By employing a Kalman filter and assuming the linearity of rails, the rail heads can be effectively tracked. However the absolute accuracy of their obtained laser point data goes up to 5 cm with only 1 cm relative accuracy. Hence, their methodology is not meant to monitor the absolute position of the rail boundaries.

Singh et al. (2019) perform edge detection on nadir UAV imagery to extract the rail lines. The imagery is preprocessed with Gaussian smoothing and converted to HSV colors in order to extract the railway area and mask the surroundings, which would not be feasible in RGB space. They apply Canny edge detection on the grey-scale image and clean up the output with a morphological operation. Using the GSD, the distance between the rails is measured directly within the rectified images. However, they do not obtain absolute measurements of rail positioning, which is what we achieve with our method.

A much more accurate assessment is performed by Ghassoun et al. (2021), where the absolute positioning of crane tracks in a harbour container terminal is determined using UAV imagery and an adjusted network of ground control points. A digital surface model is derived from the imagery from which rail edges are determined. Since an extensive ground control network is required to achieve the desired precision, this method is very intensive. Furthermore, track access is required to place the markers before the drone flight, which is not possible on railway beds.

Edge detection and line extraction To delineate the rail boundaries in UAV imagery, several approaches are possible. First, edge detection followed by line extraction is a two step process that identifies intensity changes in an image. These edges can then be used to detect straight lines. Edge detection and line extraction are still relevant topics for current research and many novel methods have been proposed. Canny edge detection (Canny, 1986) and Hough transform are commonly used to identify (rail) lines in UAV imagery (Singh et al., 2019; Khuc et al., 2020), but newer, deep learning methods could improve the accurate localisation of the rails so that gauge and positioning can be determined confidently. Holistically-Nested Edge Detection (HED) by Xie and Tu (2017) integrates holistic image training and prediction with multi-scale and multi-level feature learning to improve edge detection. The deep learning approach is an end-to-end object boundary detection system that is easily implemented. Inspired by HED and Xception networks, Soria et al. (2023) propose DexiNed. This deep learning network is trained on a dataset specifically designed for edge detection. The resulting trained net generalizes well to other domains without fine-tuning. Since HED and DexiNed are edge detection methods, lines still need to be extracted from the edge maps. For this step, we use the Hough transform (Hough, 1962). To the best of our knowledge, we are the first to apply HED and DexiNed to UAV rail imagery.

A second possibility is the use of boundary detection methods. Boundaries, in contrast to edges, are associated with objects. Segmenting the pixels based on similarity and object-recognition, can help in generating accurate boundaries which can then be used to extract Hough lines. The recently released Segment Anything Model (SAM) (Kirillov et al., 2023) uses a deep learning network trained on 11 million images. The model consists of an image encoder and prompt encoder or mask decoder depending on whether the goal is to localize a prompted object or generate a segmentation mask In this work, we use SAM's mask decoder to acquire a segmentation map of the detected objects. We are the first to apply SAM to railway monitoring.

Line segment detection Line segment detection is useful for rail localisation avoiding a multi-step process (separately performing edge/boundary detection and then line extraction). Gu et al. (2022) propose a light-weight line segment detection network: Mobile Line Segment Detection (M-LSD). A matching and geometric loss function during training enables the model to integrate geometric cues. Besides retaining performance, the model is faster than other state-of-the-art line segment detection networks.

The claim of these methods is that they do not require domain-specific fine-tuning, but their feasibility and accuracy must be tested.

3. DATA

The imagery is acquired with a DJI Matrice 300 drone equipped with a Phase One iXM-100 camera and Real Time Kinematic (RKT) Global Navigation Satellite System (GNSS) flying 26.4m above ground. The RTK GNSS receives real-time correction signals from multiple GNSS reference stations. Only nadir imagery was captured. Ground control points (GCP) were measured with a total station every 25 meters on both sides of the railbed. The considered section includes 191 images and 4 GCPs. The image size is 11664x8750 pixels. The study area is shown in Figure 2 with the as-designed rails and GCP projected over the orthomosaic.

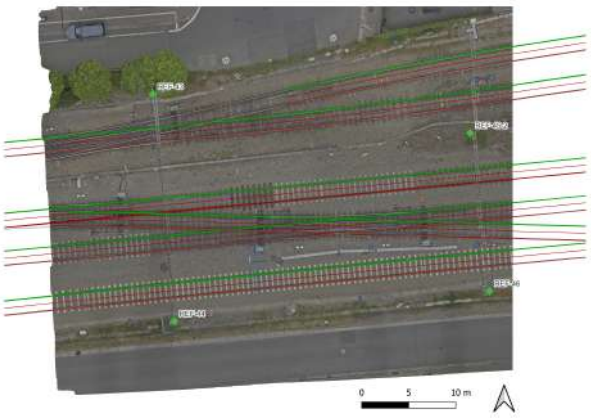

Figure 2. The study area captured by the drone. The 4 GCPs are positioned 25 m apart along the railbed. The as-designed rails are projected in green and red on the orthomosaic.

Using the mentioned parameters, the GSD for the images is 1.17 mm, which is an extremely high resolution. Figure1 shows an

example of high level of detail of the imagery captured with the drone. Due to the very fine GSD, we can accurately localize the rail edges, making it possible to calculate the gauge (distance between inner edge of the rails) directly from the undistorted images using the detected lines.

For railways with a maximum speed of 160 km/h, the local requirements for railway placement are as follows:

- Absolute positioning:
  20 mm vertical tolerance
  10 mm horizontal tolerance
- Gauge:
  -2 mm to +4 mm local gauge tolerance

It is important to note that the mentioned as-designed rails are derived from the manually annotated lines in the photogrammetric point cloud. In an ideal setup, the as-designed rail lines should be acquired from the CAD design file. In this project, no as-designed information was available. Thus, we chose to work with the manually created CAD file and consider these lines the as-designed lines. The obtained planimetric deviations in the results section are thus not accurate. However, the developed methodology is still applicable. Furthermore, the gauge deviation results are independent of the precision of the as-designed rail lines.

4. METHOD

In this section, an unsupervised pipeline is presented to detect the boundaries of the rails, measure the gauge between the detected rails, and visualize the deviations from the as-designed rail geometries. Compared to a photogrammetric point cloud, the Ground Sampling Distance (GSD) of the nadir images is superior and, thus, the planimetric assessment can be directly performed in the undistorted images.

The proposed workflow (Figure 3) to measure rail deviations in UAV imagery is divided into four steps : 1) Photogrammetric preprocessing: In the preprocessing step, the as-designed rails are reprojected onto the undistorted image in order to select relevant rail areas. 2) Rail boundary detection: in subdivided tiles, the rail boundaries are obtained via various line extraction methods. 3) Inner rail boundary pair selection: the line segments for which the distance represents the track gauge can be paired. 4) Visualisation of positioning: gauge and planimetric absolute deviation from the as-designed rails are shown in color-coded maps.

4.1 Photogrammetric preprocessing

In the preprocessing step, the as-designed rails ($q \in Q$) are projected on the imagery to mask irrelevant areas. Next, the large images are tiled; only tiles that contain as-designed rails are retained and processed in Section 4.2. This reduces required storage and computational power.

In order to project the as-designed rails given in a local projection system, the images are aligned in Agisoft Metashape. Since the drone flew with RTK GPS, coordinates of the drone camera center are obtained in the global coordinate system WGS84. We transform the coordinates to the national projection system before importing the images into Agisoft Metashape. Thus, the aligned camera poses contain the location and the orientation in the local coordinate system, and the Cartesian transformation matrices are then used to transform given world coordinates to pixel coordinates using the well-known Equation (1).

The relationship between pixel and world coordinate system can be expressed as follows:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R^T} & -\mathbf{R^T t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

where λ = scale
u, v = pixel coordinates
$\mathbf{K}$ = intrinsic camera parameters
$\mathbf{R}, \mathbf{t}$ = rotation and translation of the camera
X, Y, Z = world coordinates

Radial and tangential distortion are of importance. Therefore, we undistort the images in the software. The improvement of the undistortion on the coordinate projection is illustrated in Figure 4b. Adding GCPs further improves the alignment of the cameras leading to Figure 4c. From Agisoft Metashape, the undistorted images and an .xml file containing the camera poses of the images are exported.

4.2 Rail boundary detection

To extract the location of the rail boundaries from the tiled images generated in the preprocessing step, we experiment with the five methods listed below.

- Canny: Canny edge detection & Hough transform
- HED: Holistically-nested edge detection & Hough transform
- DexiNed: DexiNed edge detection & Hough transform
- SAM: Segment Anything Model & Canny edge detection & Hough transform
- M-LSD: Mobile Line Segment Detection

The first four multi-step methods extract edges or boundaries which can then be used as input to a Hough line transformation whereas the last one-step method (M-LSD) directly detects lines. We refer to Figure 5 for a detailed depiction of the performed steps involved in the five methods.

Edge detection Canny, HED, and DexiNed are edge detection algorithms. Canny is threshold based, while HED and DexiNed are deep learning models. To extract lines from the generated edges, the Hough transform is used. However, the edges generated by HED and DexiNed require some processing before the Hough transform can be used. The output first needs to be converted to a binary image using a threshold value. In our experiments, we used a pixel value of 128. The resulting edges are then eroded to obtained thinner edges. Then, the Hough transform algrithm is used to obtain lines.

Boundary detection Furthermore, we explore boundary detection using the Segment Anything Model (SAM). By applying SAM, we generate object masks for all detected objects in the images, including the rails. To obtain a boundary map from the segmentation masks, we combine all the masks into a single layer, assigning a unique pixel value ID to each object instance (in Figure 5, we call this instancing) and obtain a single band image. The pixel values range from 0 to the number of detected instances. Canny edge detection is then applied to this boundary

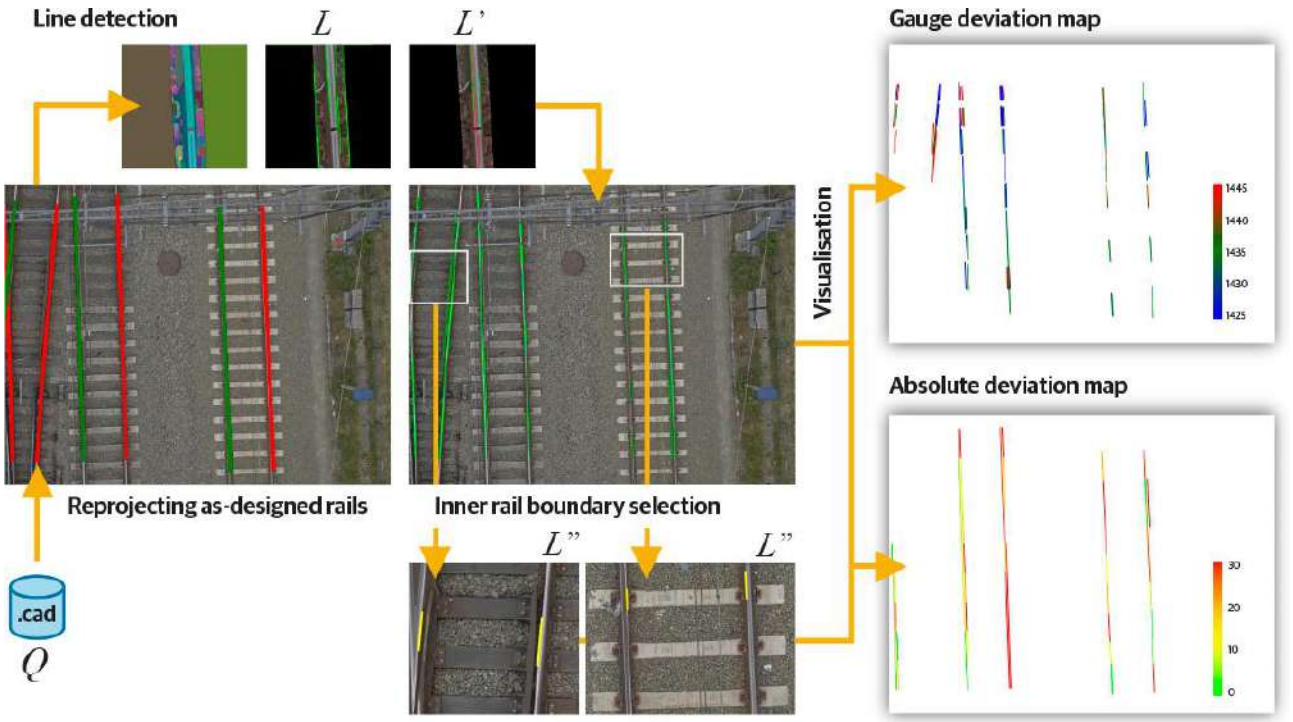

Figure 3. The proposed workflow starts by reprojecting the as-designed rail lines Q onto the undistorted images. The rail boundaries L' are extracted using Segment Anything Model (SAM) and filtering out false positives from all detected lines L. The inner rail boundary lines can be paired in L'' to obtain gauge and absolute deviation maps.

(a) Image captured by drone, alignment without GCP

(b) Undistorted image

(c) Undistorted image, alignment with GCP

Figure 4. Projection of a GCP (a) in the original image captured by the drone, (b) after undistorting the captured image, and (c) adding the GCP to the alignment in Agisoft Metashape.

map. Since Canny detects edges based on the gradient between pixel values and edges need to be detected between each instance, we set the lower Canny threshold to the value of 0.5 so that edges between each object instance are detected. Finally, the Hough Transform algorithm is used to extract straight lines from the resulting Canny edge map.

Line segment detection In the final approach, we directly perform line segment detection without employing Canny or Hough line transformation. This is achieved using the Mobile Line Segment Detection method (M-LSD), which extracts line segments from the images in a single step.

Line filtering The drawback of masking the irrelevant areas around the rails, as described in Section 4.1, is that the edge of the mask also results in a detected line. Furthermore, lines that are not in the same direction as the as-designed rails, can already be removed. False positives can, as shown in Figure 6, be filtered out based on two conditions: the detected line l must be nearly parallel to at least one as-designed rail line q and the distance between l and the nearest q must be within the buffer zone. The

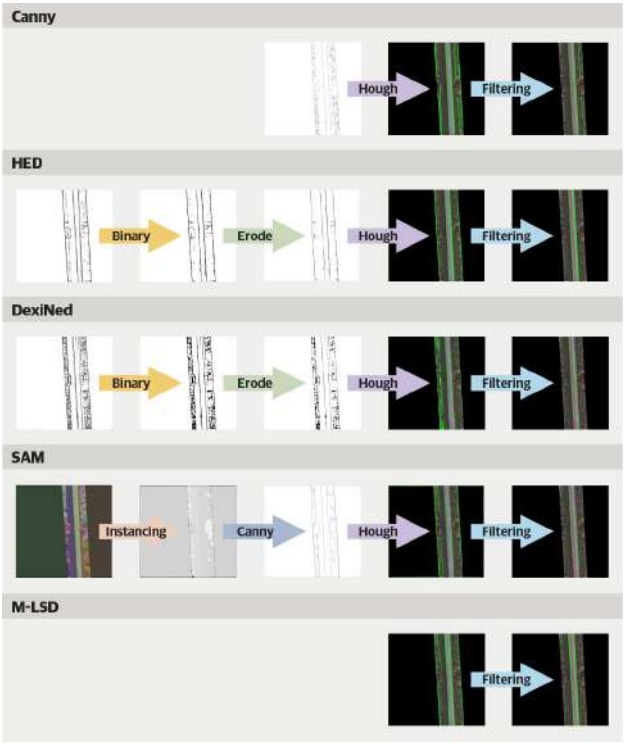

Figure 5. The required processing steps for the rail boundary detection methods.

resulting set of filtered lines L' is defined in Equation (2) where the distance and angle between l and q are below thresholds t_B and t_α:

$$L' = \{l \in L \mid \exists q \in Q, d_{max}(l,q) \leq t_B \,\&\, \angle(l,q) \leq t_\alpha\} \quad (2)$$

where d_{max} is defined as the maximum distance between the endpoints of l and the nearest as-designed rail line q, as in Equation (3) in which P_1, P_2 are the endpoints of detected line l.

$$d_{max}(l,q) = \max\left(d(P_1,q), d(P_2,q)\right) \quad (3)$$

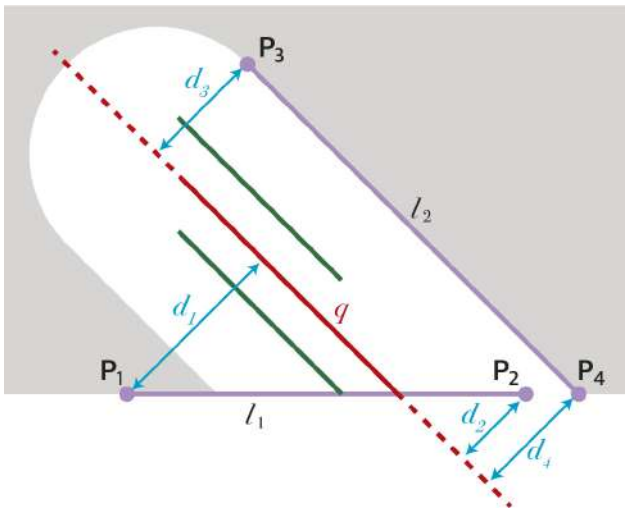

Figure 6. False positives are removed using the maximum distance from the endpoints of the detected lines l to the nearest as-designed rail line q.

4.3 Inner rail boundary selection

In step 3, the inner rail boundary lines are selected as pairs between which the gauge can be measured. We define L'' as the set of pairs of lines from L', in which each rail line is matched with another rail line in order to determine the gauge, using several constraints, as defined in Equation (4):

$$L'' = \{(l_1,l_2) \in L' \mid \exists q \in Q, |d(l_1,l_2) - G| \leq t_G \,\& \quad d(l_1,q) \leq t_d \,\&\, d(l_2,q) \leq t_d\} \quad (4)$$

where l_1, l_2 = inner rail boundary pairs
G = the expected gauge
t_G = maximum gauge offset
t_d = maximum deviation tolerance

When determining the distance between line segments l_1 and l_2, we only want to take into account the overlapping part. Considering the end points of both segments and constructing their projection point on the other line, either none of two of these projections lie within the other segment. We only consider the latter and take the average of the distances as illustrated in Figure 7 and defined in Equation (5) where P_2 and P_3 are the line endpoints that have a projection point on the other line segment.

$$d(l_1,l_2) = \frac{d(P_3,l_1) + d(P_2,l_2)}{2} \quad (5)$$

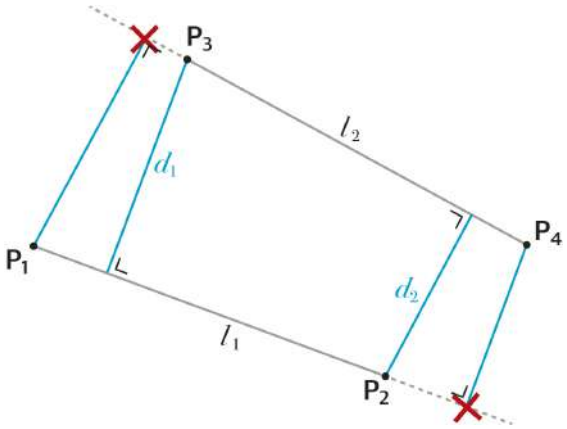

Figure 7. The distance between detected rail lines l_1 and l_2 is computed as the average of the distances between end points and their projection on the other line, if this projection lies on the other segment.

4.4 Gauge and deviation maps

Our proposed workflow concludes with a visualisation of the deviation from the expected gauge and deviation from the as-designed rail boundary lines. Each individual rail must be within a tolerance of the as-designed plan. To determine whether their location is within this tolerance, the coordinates of the detected lines can be transformed from pixel space to world space. The lines can be imported in the as-designed CAD file where it is straightforward to determine the distance between the as-designed and as-is rails. But, within the image directly, we can also calculate their offset from the matching as-designed rail line. This is done for all the lines in L''.

5. RESULTS

5.1 Edge Detection and Line Extraction

We compare the Canny edge baseline to HED, DexiNed, SAM, and M-LSD. Some factors to consider for the evaluation are the detection of the rail boundaries, the number of lines detected, and the processing speed. In Figure 8, we show some examples of the detected lines using the described methods. The detected lines by various algorithms are shown in green and the as-designed inner rail boundary is shown in red. Figure 8 a) shows a simple single rail. Only Canny fails to detect both rail boundaries (inner & outer). In Figure 8 b) a crossing is shown. Figure 8 c) and d) show cabled crossing perpendicular are diagonally over the rail. The filtering using Equation (2) effectively removed most detected lines on the mask edge and lines not parallel to the as-designed line.

We find SAM gives very little false positives but requires intensive processing. M-LSD and HED, on the other hand, generate overlapping lines, but give quick results. A fair balance can be found with DexiNed, many short lines are detected without overlap and processing speeds are significantly faster than SAM.

5.2 Determining Gauge & Positioning Deviation

The resulting gauge and positioning deviation can be shown in a color-coded deviation map as shown in Figure 9. For each matching line pair the average distance is calculated. The standard gauge is 1435 mm. As described in Section 3, the maximum

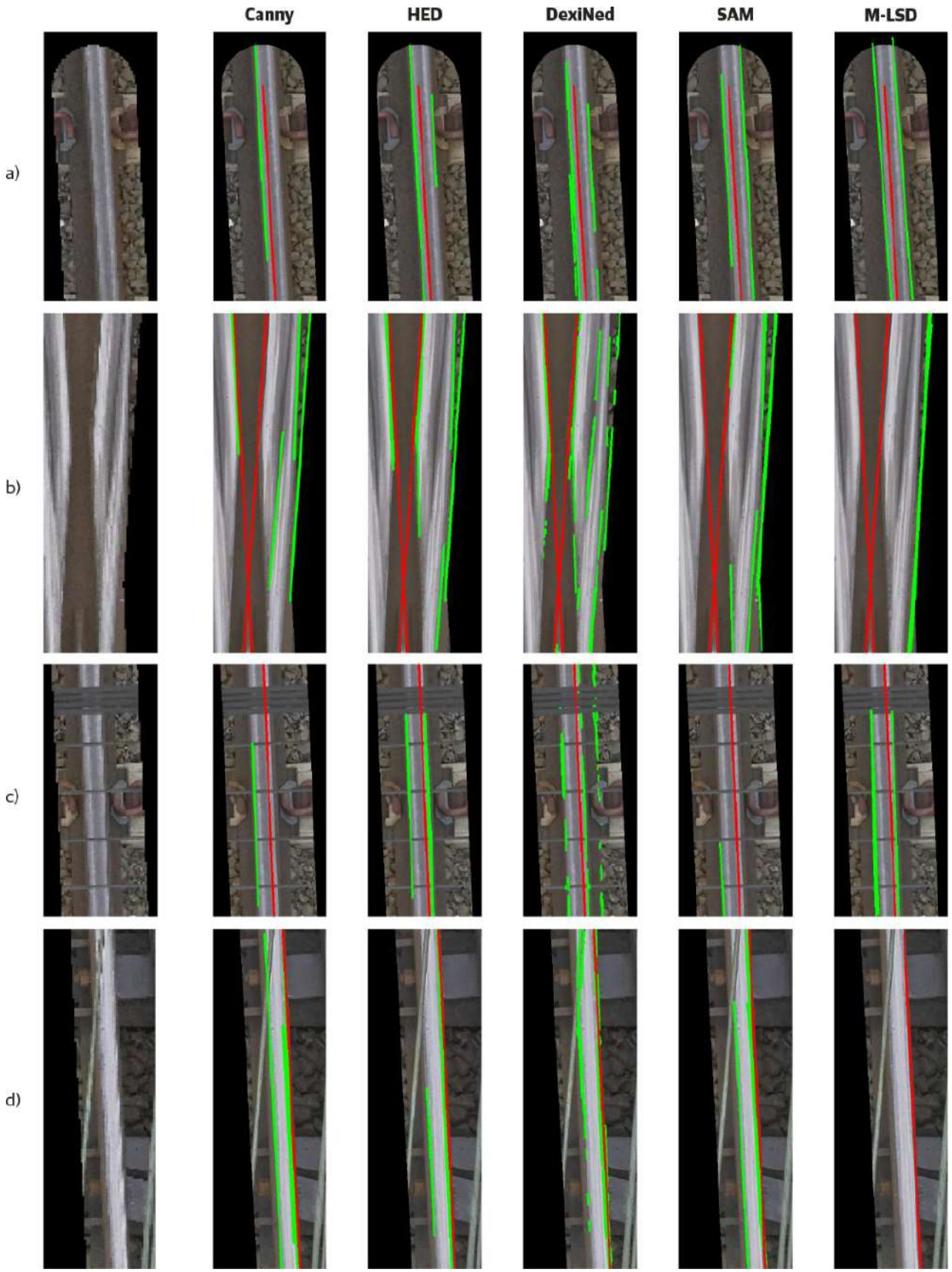

Figure 8. We show the results for the tested methods for a) a straight rail, b) a crossing, c) perpendicular visual obstructions of a rail, and d) a diagonal overhead line creating visual obstruction of a rail. From left to right: Canny edge detection with Hough lines, Holistically-Nested Edge Detection (HED) with Hough line extraction, DexiNed edge detection with Hough line extraction, Segment Anything Model (SAM) with Canny edge detection and Hough lines,and Mobile Line Segment Detection (M-LSD).

gauge deviation tolerance is -2 mm to +4 mm. Based on its deviation from the expected gauge, the line pairs are colored in red for a gauge that is 1440 mm or more or in blue for a gauge that is 1430 mm or less. For the deviation for the absolute positioning, a similar color coding is utilised as shown in Figure10.

The matching lines in L'' are filtered on a distance close to the expected gauge, as defined by equation (4). However, some false positives still occur as shown in Figure 11.

6. DISCUSSION

From our experiments we find that the tested newer edge detection and line extraction algorithms outperform the traditional Canny edge detector. However, obstructions and uneven wear of the rails makes it visually difficult to determine the rail edge location. The generated deviation maps are thus incomplete, with gaps present, but for each detected rail pair, the deviation can accurately be determined. Our method has the advantage that localisation accuracy can be improved with an improved ground control network as done by Ghassoun et al. (2021).

Since no accurate ground truth as-designed rail lines were available, the rail boundary detection methods could not be quantitatively evaluated. A new campaign is planned in which the ground truth will be measured with a rail measurement trolley. The accuracy of the detected rail boundaries can then be verified. Furthermore, with foreseen oblique image capturing, the rail localization can be extended to a 3D evaluation. The altimetric assessment can then be computed from the photogrammetric point cloud. Lidar data, as employed by Ariyachandra and Brilakis (2020), would also allow for the altimetric evaluation. However, the drone cannot fly with both optical and Lidar sensor, and Lidar data would only be sufficient to determine altitude, and not planimetric rail boundary localisation.

7. CONCLUSION

In this work, we present a UAV-based rail boundary localization method for gauge and absolute positioning deviation measurement. Drone inspections for railways are a promising alternative to the traditional measurement trolleys, which are not only time-intensive but also dangerous as rail access is required. In this work, we explore a remote sensing method using RTK drone imagery and GCPs. The local government imposes strict requirements on the gauge deviation and on the absolute positioning of the rails. Our workflow consists of preprocessing where cameras are aligned in Agisoft Metashape so that as-designed rail edges can be reprojected onto the imagery. Then, five algorithms are tested to detect the inside rail boundary edges. Specifically, we compare newer deep-learning methods to the well-known Canny edge detector. False positive detections are filtered out using angle and distance measurements from the as-designed rail lines. The gauge is measured between matching rail boundary pairs and the deviation from the as-designed rails is mapped using color codes. We find that among the tested line extraction methods, DexiNed is a good trade off between detections, speed, and minimal overlapping lines. SAM is a good method to reduce false positive detections, but requires powerful processors. Our developed method effectively detects inner rail boundary lines and results in easily interpretable deviation maps.

Although there is no doubt that a rail measurement trolley will provide more accurate results, traditional methods are not an

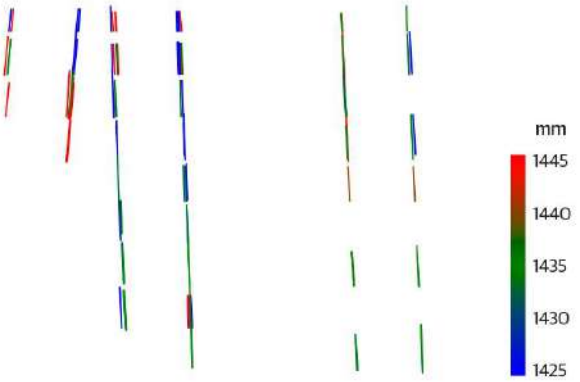

Figure 9. The gauge deviation map is a visualization of the detected rail line pairs colored based on their deviation from the expected gauge. This example shows M-LSD lines.

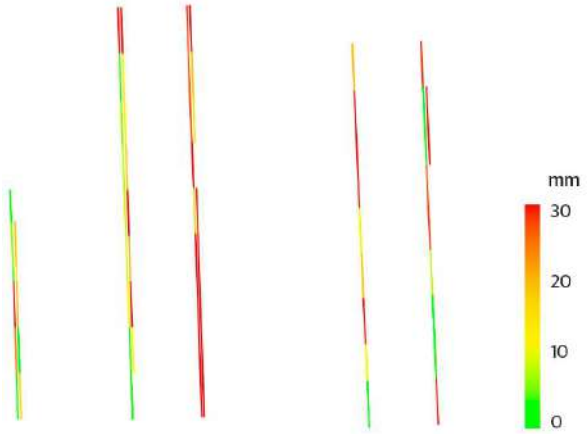

Figure 10. The planimetric deviation from the as-designed rail lines can be visualised using a color map. This example shows M-LSD lines.

Figure 11. A False positive inner rail pair. The detected line on the left is an outer rail edge. It is matched with the bottom of the rail on the right. The distance between the two is equal to the gauge.

option when rail access is denied. UAV monitoring has various benefits, not only is it safe and fast because no track access is required, it is also cheaper.

References

Ariyachandra, M. R. M. F., Brilakis, I., 2020. Detection of Railway Masts in Airborne LiDAR Data. 146(9), 04020105. https://ascelibrary.org/doi/10.1061/%28ASCE%29CO.1943-7862.0001894.

Canny, J., 1986. A Computational Approach to Edge Detection. PAMI-8(6), 679–698. https://ieeexplore.ieee.org/document/4767851.

Ghassoun, Y., Gerke, M., Khedar, Y., Backhaus, J., Bobbe, M., Meissner, H., Tiwary, P. K., Heyen, R., 2021. Implementation and Validation of a High Accuracy UAV-Photogrammetry Based Rail Track Inspection System. 13(3), 384. https://www.mdpi.com/2072-4292/13/3/384.

Gu, G., Ko, B., Go, S., Lee, S.-H., Lee, J., Shin, M., 2022. Towards light-weight and real-time line segment detection.

Hough, P. V., 1962. METHOD AND MEANS FOR RECOGNIZING COMPLEX PATTERNS.

Jwa, Y., Sonh, G., 2015. Kalman Filter Based Railway Tracking from Mobile Lidar Data. II-3/W5, 159–164. https://isprs-annals.copernicus.org/articles/II-3-W5/159/2015/.

Khuc, T., Nguyen, T. A., Dao, H., Catbas, F. N., 2020. Swaying displacement measurement for structural monitoring using computer vision and an unmanned aerial vehicle. 159, 107769. https://linkinghub.elsevier.com/retrieve/pii/S0263224120303079.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., Girshick, R., 2023. Segment Anything. https://arxiv.org/abs/2304.02643. Publisher: arXiv Version Number: 1.

Singh, A. K., Swarup, A., Agarwal, A., Singh, D., 2019. Vision based rail track extraction and monitoring through drone imagery. 5(4), 250–255. https://linkinghub.elsevier.com/retrieve/pii/S2405959517302886.

Soria, X., Sappa, A., Humanante, P., Akbarinia, A., 2023. Dense extreme inception network for edge detection. 139, 109461. https://linkinghub.elsevier.com/retrieve/pii/S0031320323001619.

Xie, S., Tu, Z., 2017. Holistically-Nested Edge Detection. 125(1), 3–18. http://link.springer.com/10.1007/s11263-017-1004-z.