

RAILWAY DETECTION: FROM FILTERING TO SEGMENTATION NETWORKS

B. Le Saux, A. Beaupère, A. Boulch

DTIS, ONERA, University Paris Saclay,
F-91123 Palaiseau, France

J. Brossard, A. Manier, G. Villemin

ALTAMETRIS
FR-75010 Paris, France

ABSTRACT

This paper deals with classification of remote sensing data to extract objects for industrial mapping. While land-cover or urban mapping have been extensively studied, industrial cartography remains a field yet to explore, in spite of tremendous needs. We present and compare here four approaches for railway detection in very high resolution images. They use various kind of filtering approaches, including the trained filters of fully convolutional networks. Moreover, they benefit from different a-priori and post-processing techniques to make them more robust. We evaluate all approaches on a challenging dataset captured on an operating station site with complex objects.

Index Terms— Rail detection, Railway detection, Filtering, Neural Networks

1. INTRODUCTION

With the development of new, practical means such as Unmanned Aerial Vehicles (UAVs), monitoring of industrial sites by remote sensing is now a reality. Remote sensing has been adopted in various industries such as oil and gas, mining or transports. In this latter field, it may answer to a crucial need of railway operators: ensuring the optimal safety of the passengers. Indeed, the knowledge of the current operational state of the infrastructure is essential to network administrators [1].

The recent use of drones in this sector allows a significant increase in information gathering efficiency, notably by providing in good time extremely high resolution aerial images of railway furnitures. Currently, these images are used as a visual support for manually marking relevant elements of the infrastructure (rails, speed beacons, overhead line poles and catenaries, etc.), aggregated as a normalized blueprint of the railroad studied division. Automating image recognition procedures for cartography stands as a crucial optimization in this sphere.

Most railway detection works benefit from the newly available sensing capacities. The measurement precision is crucial, and first works relied on airborne LiDAR data to extract railroad objects [2]. At that time, it was considered that

the enormous amount of image data and radiometric variations impeded the use of imagery. More recently, approaches to process very high resolution imagery were proposed: in [3], detection is performed through histogram of oriented gradients and support-vector machine learning, mostly applied to oblique images.

This paper is organized as follows. In Section 2 we present how data are acquired and pre-processed. We describe four approaches for rail detection in Section 3. We assess and compare these approaches on a challenging dataset which corresponds to an operational use case in Section 4. We present our concluding remarks in Section 5.

2. DATA ACQUISITION

In the following, we present the procedures for data acquisition and geometric processing which allow to build orthorectified images of a railway station. They were established by SNCF, the French railway operator, and are deployed in railway cartography projects, such as the one which lead to the Rennes dataset used in the experiments of Section 4.

Data acquisition. The raw images are collected with a Sony Alpha NEX-7 camera, embedded on a light multi-rotor drone performing at an average altitude of 50 m above ground level. The internal pixel size of the camera is 3.9 μm and renders 6000 \times 4000 RGB, 8-bit-depth-per-band images, with a 4:2:0 chrominance compression format. Each frame is tagged with Global Positioning System (GPS) and inertial measurements of the drone at the instant of shooting. Ground Control Points (GCPs) are physically set on the studied scene and their geolocation is manually obtained by means of a high precision GPS pole. GCPs are used to improve the precision and accuracy of the produced data.

Data processing. A 3-dimensional scene reconstruction based on the GCPs and points automatically matched by a SIFT algorithm is computed using the inertial and GPS values with the Pix4D software¹. Then, the ground surface is modeled using multi-scale geometric and color features [4]. The final ortho-rectified image is obtained by projecting retroactively segmented ground pixels from the original frames onto the terrain model, allowing a vertical rendering at any point.

¹<https://pix4d.com/>

This process drastically reduces perspective errors and rail obstruction hazards. The parameters used during acquisition and processing ensure a 0.011 m/pixel final resolution.

Cartography and ground truth. The ortho-rectified image is used by image analysts to manually draw maps of the railway station, which include platforms, signals, buildings and above all, rails. These vector maps follow the SNCF-developed EPURE norm, which implies that rails are represented as segmented lines anchored to points located on the center of the rail width every 10 m. This standard does not take rail curvature into account to avoid interpolation in railway track studies and analysis. Consequently, the real ground-truth is defined as the geo-located anchor points, an not as the whole map.

3. RAIL DETECTION ALGORITHMS

In the following we present the four approaches for rail detection which we will compare in Section 4.

User aided correlation matching. The first approach is a looped cross correlation matching process. The process is the following. A user provides an oriented rectangular sample of a rail in the image. The direction parameter serves in establishing an oriented triangular search area with a top apex angle of $2 \times 11^\circ$, consistent with rail curvature. Several matches are found in the area but only the nearest to the original is retained if it gets a correlation score over 0.4. The new found oriented rail segment becomes the next matching pattern and the process is repeated until no further match is found. Using the previous user input, the algorithm restarts from the beginning using the opposite direction.

This approach allows to drastically reduce the time for manual extraction of rails in aerial images, but nevertheless requires user inputs for each railway.

Line detection with conditional morphological and orientation filters. In the second approach we use various morphological and orientation filters to extract rails (cf. Fig. 1). To reduce false detections, we first eliminate shadows from the images using Normalized Difference Index as described by Singh *et al.* [5]. Knowledge of pixel size allows us to apply a differential Top-Hat transformation [6] using a disc as a structuring element with a diameter corresponding to the rail width, increasing the relevance of the filter. Lines are detected on the resulting image using the curved-line extraction algorithm described in [7].

Two kinds of constraints are introduced to improve the results through post-processing. First we use the French railway Geographic Referential of Infrastructure (GRI) to get a global orientation of the tracks. We compare it to the resulting lines and thus eliminate mis-oriented results. A set of morphological operations (i.e. dilation, opening, union, closing) is then applied on the candidate regions to concatenate the results. Second, using the knowledge that each rail is located 1.5 m away from its sibling, a discriminant parallelism and distance

analysis is finally applied.

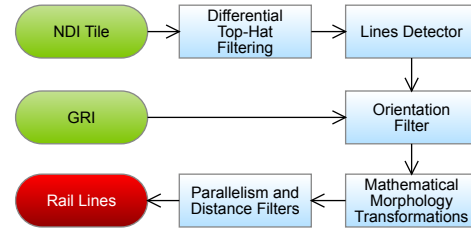


Fig. 1. Workflow of the line detector based on conditional morphological and orientation filters.

Line detection with Touzi filters. Touzi filters [8] are commonly used for edge extraction in extremely noisy environments, originally introduced for speckle filtering in radar images. Given an orientation a priori (available from the GRI for example), the filter computes a radiometry ratio between rectangular areas on both sides of the chosen direction. By thresholding the resulting map, pixels which belong to strong, long-enough edges are extracted.

In a second time, a vectorization process is performed by applying a step-by-step search along the direction of the detected edges. It allows to get rid of mid-detections due to wires or pylons.

Semantic Segmentation Networks. In this last approach, we use a Convolutional Neural Network (CNN) designed for supervised, semantic segmentation. CNNs are feed-forward neural networks comprised of learnable convolution kernels. Those filters perform feature extraction in order to build an internal, abstract representation of the input, optimized for later classification. We use a standard fully convolutional network (FCN) with an encoder-decoder structure: SegNet [9]. FCNs usually get RGB images as inputs and infer structured dense predictions by assigning a semantic class to every pixel of the image. We follow the approach of Audebert *et al.* [10, 11] defined for aerial image segmentation.

CNNs are supervised algorithms which require training data. Manually designed maps described in Sec. 2 are used to generate training rasters at the image resolution. They comprise a range of objects which can be extracted from the vector maps such as rails, buildings, electrical equipment, etc. However they are often inconsistent: more than 90% of the image points remain unlabeled, some classes (e.g. buildings) are partially annotated, and some classes only have a few samples while others are better represented.

The encoder part of the SegNet is initialized with pre-trained VGG-16 weights, while the decoder part is randomly set. It is trained on 128×128 patches with stochastic gradient descent. Moreover, to deal with the class imbalance problem, partial loss of class c in the cross-entropy function is weighted by $\frac{\text{med}_{1 \leq i \leq c}(\text{freq}_i)}{\text{freq}_c}$ for all c .

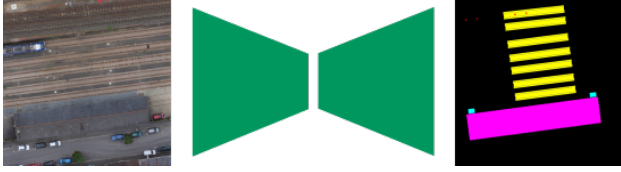


Fig. 2. Fully-convolutional network for railway detection and training data. At test phase the encode-decoder network gets unknown images and predict a segmentation map.

4. RAILWAY DETECTION RESULTS

The Rennes dataset used in this study was acquired over the railway station of Rennes, France, in 2017, following the procedure described in Section 2. For processing purposes, it was segmented in 5000×5000 ortho-rectified images at resolution of 2cm per pixel. It comprises 34 images for training (i.e setting hyper-parameters of the algorithms or actually training the segmentation network) and 21 images for testing which were not seen before the evaluation (see Fig. 3-a for some examples).

The test area implies 1172 rail control points (RCPs) (cf. 2) which establish a consistent common testing set. The ground-truth for evaluation comprises those RCPs and a small neighborhood around them (12 pixels) for taking into account the width of rails, which lead to more than 14k pixels. This allows to measure the localization precision of the detection with respect to the rail: centered or on the edge. The various rail detection algorithms were evaluated according to the producer accuracy $Acc. = \frac{T}{|\mathcal{P}|}$ where $|\mathcal{P}|$ is the number of rail pixels in the ground-truth, and T is the number of true positives i.e the number of points that received the "rail" label. These accuracy results are summarized in Table 1. Besides the overall results on the whole test set (denoted by *all test*) which also contains neighboring urban areas with buildings and roads, we also focus on a smaller set of 3 images with mostly railways (hereafter the *rail subset*) which corresponds to roughly 200 RCPs and 2.4k pixels. This allows to compare the rail extraction methods when no other salient, distracting features are present in the images, as would be the case for example if a mask obtained from open-source maps was applied as a preprocessing step.

In Fig. 3 we show rail detection results obtained by all methods along the corresponding tile. First, it appears that all approaches are quite successful at detecting rails: all retrieve the railways in the upper part of the images while having more-or-less false positives in the neighbor station or city buildings. At second sight, differences appear. As predictable, the user-initialized correlation matching (Fig. 3-b) yields no false alarms, and most rails are found by the user. Mainly, errors occur when the step-by-step tracking of the rail fails. Conditional morphological and orientation filters (Fig. 3-c) obtain good railway outlines (i.e. pair of rails)

Table 1. Railway detection accuracies for methods of Sec. 3, computed on the whole test set (21 images) and on a subset with mostly railways (3 images).

Method	All test (14k samples)	Rail subset (2.4k samples)
User corr. matching	79.67	76.87%
Conditional filter	75.40%	71.62%
Touzi filter	76.66%	92.74%
Segmentation network	50.38%	89.00%

thanks to the strong a priori about orientation or parallelism that is introduced in the approach. A few false alarms occur in city environments, and the false negatives are mostly due to the parallelism a priori: some rail may be discarded if no companion rail appears due to image default or vegetation. Touzi filters (Fig. 3-d) use only orientation a priori and follow a step-by-step process for vectorization of rail detection. They get extremely good results, and do not miss much rails. However, this also the approach which leads to the most numerous false alarms, typically on buildings and road edges which have the same orientation as the a priori. Finally, the fully-convolutional segmentation network (Fig. 3-e) is the most brute-force approach. Indeed, no a priori is given apart from the training data, and no post-processing is performed to obtain continuous rail lines (vectorization process). However it performs remarkably well: most railways are retrieved as seen in the detection maps, and false alarms are structurally different from those of conditional or Touzi filters (no line effect) so they are easier to spot. The main problem is the *dashed line* effect which occurs if looking closer and appears when a rail is rusty (e.g. for secondary tracks) or occluded by wires or vegetations. Overall, it appears that adaptability (to adaptation or local appearance) is a feature to care about when designing a railway detection algorithm.

Table 1 summarizes the accuracy measures. Considering all 21 test images with city surroundings, the filtering approaches obtain good results with accuracies around 75% or 80%. However conditional or Touzi filters do not require user supervision at test which is an advantage for large scale deployment. The segmentation network gets mixed results since only half of the control points are detected due to the lack of constrained continuity in the detection. If we focus on the rail subset however, Touzi filtering and the segmentation network reach around 90% accuracy. This hints that a priori constraints as simple as an infrastructure ground coverage mask are a promising way to improve the performances.

5. CONCLUSION

In this paper we have presented 4 filter-based approaches for rail detection. They differ by the nature of the filters, which can even be learned from a dataset in the case of a fully-

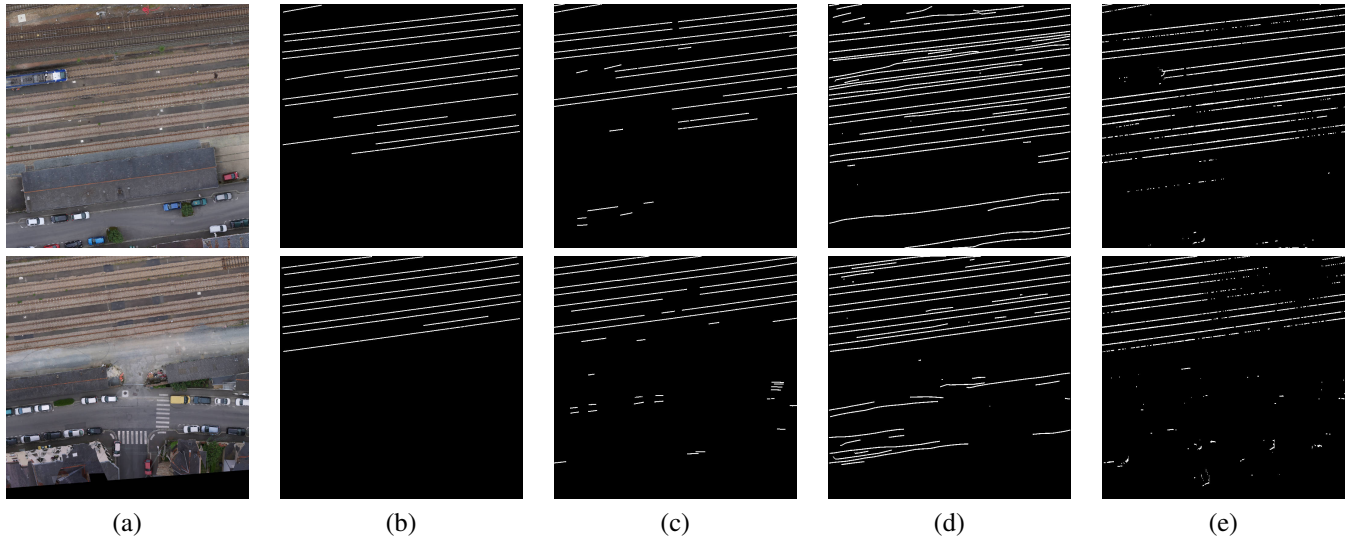


Fig. 3. Railway detection results (in blue) with methods of Sec. 3: (a) Original image; (b) User correlation matching; (c) conditional filter; (d) Touzi filter; (e) Segmentation network.

convolutional neural network, and by the amount of a priori which is introduced, including user supervision, geographical and thematic constraints, or no a priori. All approaches lead to more-or-less satisfying results depending on the use case and the available resources. If aiming at the most automated approach, user supervision is not required since 3 other methods got equivalent or better results, and orientation and parallelism are not necessary, since they can be learned from the data. However the infrastructure coverage and vectorization by means of a step-by-step line building procedure help predict better detection maps.

Thus, by taking the best of each approach, this benchmark shows it is possible to design an automated, even more efficient process for rail detection. A promising path consists in mixing simple a priori constraints into the fully-convolutional segmentation networks.

6. REFERENCES

- [1] Vassilios A Profillidis, *Railway management and engineering*, Ashgate Publishing, Ltd., 2014.
- [2] M Neubert, R Hecht, C Gedrange, M Trommler, H Herold, T Krüger, and F Brimmer, “Extraction of railroad objects from very high resolution helicopter-borne lidar and ortho-image data,” *Int Arch Photogramm Remote Sens Spat Inf Sci*, vol. 38, pp. 25–30, 2008.
- [3] J. Espino and B. Stanciulescu, “Turnout detection and classification using a modified hog and template matching,” in *Intelligent Transportation Systems-(ITSC)*, 2013.
- [4] C Becker, N Häni, E Rosinskaya, E d’Angelo, and C Strecha, “Classification of aerial photogrammetric 3d point clouds,” *ISPRS Annals of Photogram, Rem. Sens. & Spatial Inf. Sciences*, vol. 4, 2017.
- [5] K. Singh, K. Pal, and M.J. Nigam, “Shadow detection and removal from remote sensing images using ndi and morphological operators,” *Int. journal of computer applications*, vol. 42, no. 10, 2012.
- [6] Allan G Hanbury and Jean Serra, “Morphological operators on the unit circle,” *IEEE Transactions on Image Processing*, vol. 10, no. 12, pp. 1842–1850, 2001.
- [7] Carsten Steger, “Extraction of curved lines from images,” in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on.* IEEE, 1996, vol. 2.
- [8] R. Touzi, A. Lopes, and P. Bousquet, “A statistical and geometrical edge detector for sar images,” *IEEE Trans. Geoscience and Remote Sensing*, vol. 26, no. 6, 1988.
- [9] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” *arXiv preprint arXiv:1511.00561*, 2015.
- [10] Nicolas Audebert, Bertrand Le Saux, and Sbastien Lefvre, “Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-scale Deep Networks,” in *Computer Vision ACCV 2016*, Nov. 2016.
- [11] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre, “Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images,” *Remote Sensing*, vol. 9, no. 4, Apr. 2017.