Abstract

For safety purpose, railroad tracks must be inspected regularly for defects or other design non-compliances. One crucial building block in an automatic inspection system is to detect different types of railroad track objects. We introduce a novel global optimization framework to combine evidence from multiple cameras and the distance measuring instrument to improve rail object detection. Our framework leverages the cross-object spatial constraints enforced by the sequential structure of rail tracks, as well as the cross-frame and cross-view constraints in camera streams. Experimental results on real rail track-driving data demonstrates that our approach achieves superior performance compared to processing each data stream independently. We argue that our approach can be extended to other embodiments involving linear sequential structures, such as pipeline, highway and road inspection.

1 Introduction

According to a recent report by the Federal Railroad Administration (FRA), rail defects result in thousands of derailments causing casualties and a cost of hundreds of millions dollars each year. Recently, automatic vision-based rail inspection systems have shown more efficiency and reliable performance than human inspectors. Rail inspection includes a wide variety of tasks, ranging from assessing condition of different rail objects (tie plates, ties, anchors, etc) to evaluating rail alignments, surfaces and curvatures, to detecting sequence-level track defects [2, 5, 8, 11, 3, 1, 10]. Among these tasks, detecting and locating rail objects is arguably the most crucial step. However, it is quite a challenging task in the real environment. Single-frame object detection is solely based on visual information, therefore suffers from different confounding issues such as shadows, occlusion, lighting changes, change in camera view point, so on (see Figure 2).

In this paper, we introduce a novel approach to improve object detection by leveraging contextual information such as cross-frame, cross-view and cross-object constraints, making it much more robust than using visual information alone. We devise a unified framework for rail object detection and association, through global optimization, which integrates evidence at multiple frames, multiple views and multiple objects. Our framework enforces three types of constraints:

**Cross-frame constraints:** Since the same object may be observed in more than 1 consecutive video frames, we leverage the spatial constraints of detections between frames in each single camera stream.

**Cross-view constraints:** Our system uses 4 synchronized cameras with overlapping fields of view and fixed calibration parameters. Since each rail object can be observed by multiple cameras simultaneously, it is natural to exploit the spatial constraints between camera views to improve the detection confidence of that object.

**Cross-object constraints:** Rail tracks are sequential structures formed by a sequence of different objects, installed by a specific design. Therefore, the spatial layout between objects are strongly enforced by design, rather than loosely as in the general context of object detection [4, 7]. For example, the spacing between consecutive ties in a rail track should be around a constant.

![Figure 1](image-url) Overall architecture of our rail inspection system.
The building blocks of our rail inspection system include: (1) 4 synchronized cameras with overlapping views and fixed calibration parameters, mounted on the vehicle, (2) distance measuring instrument (DMI), which captures the speed of the vehicle, (3) A built-in rail component detector that takes as input the video stream from a camera and return one or more detections within each video frame, each detection associated with a confidence score. (see Figure 1)

The object association and optimization module receives 4 streams of detections and the DMI output, and through global optimization, will modify the detection decisions in each frame in the context of other detections in other frames and other cameras. The output of this module is an improved set of object detections, which may then be fed to the subsequent steps of track defect analysis and identification.

![Figure 2](image)

**Figure 2.** (a) The four cameras mounted on a hi-rail vehicle and (b) their fields of view (c) Tie plate appearance has a high variability coming from diversities in shape, size, camera view point, occlusion and light condition.

Experimental results on real rail track-driving data demonstrates our superior performance in the context of rail inspection. We also believe that our approach can be extended to other embodiments involving linear sequential structures, such as pipeline, highway and road inspection [9, 6].

## 2 Our Approach

### 2.1 Optimization Formulation

Given 4 streams \{S_1, ..., S_4\} of object states, each is the result of applying the object detection module to one of the camera streams for a duration of \(T\). Each \(S_k\) consists of a sequence of object states \(\{s^1_k, ..., s^T_k\}\). Without loss of generality, we assume that there is only at most 1 object state per frame. Our approach can directly be applied to the case where there are multiple object states per frame. Figure 3 illustrates our optimization problem. Each column in the graph corresponds to a video frame, each row corresponds to a camera view. Round nodes corresponds to object states (results of the object detector). Note that the detector may find multiple detections per frame, which results in having multiple states per frame.

We want to find the path from time 1 to time \(T\), by selecting a set of states \(S^* = \{s^1_k, ..., s^T_k\}\) optimizing the following energy function:

\[
S^* = \mathop{\text{arg max}}_{S} E = \sum_{t} \psi(s^t_k) \phi(s^t_k, s^{t+1}_k) \tag{1}
\]

where \(\psi(s^t_k)\) is the unary potential of an object state \(s^t_k\), \(\phi(s^t_k, s^{t+1}_k)\) is the cross-frame spatial constraints, \(s^t_k\) is the object state at node \((k, t)\), which initially is the input object detection.

**Cross-view Constraints:** model the spatial constraints of different object states between different camera views. We assume all camera calibration parameters are fixed, which is true in our case. Given an object state \(s^t_k\) at view \(k\), the state \(s^t_j\) at view \(l\) follows a Gaussian distribution. We represent this cross-view constraint as follows:

\[
T(s^t_k, s^t_j) = \max \left( \frac{\mathcal{N}(|s^t_k - s^t_j|; \theta_{kl})}{\mathcal{N}(|s^t_k - s^t_k + \epsilon|; \theta_{kl})} \right) \tag{2}
\]

where \(\theta_{kl} = [\mu_{kl}(k, l), \Sigma_{kl}(k, l)]\), \(\mu_{kl}\) is a 4x4 matrix of mean values, \(\Sigma_{kl}\) is a 4x4 covariance matrix, \(\epsilon\) represents the tie spacing constant, which is used in the case where the two object states \(s^t_k\) and \(s^t_j\) do not correspond to the same physical object, but the adjacent object in the sequence. \(\theta\) and \(\epsilon\) can all be learnt from labeled training data.

Now the unary potential \(\psi(s^t_k)\) is defined as:

\[
\psi(s^t_k) = f(s^t_k) \prod_{l \neq k} T(s^t_k, s^t_l) \tag{3}
\]
where \( f(s^t_k) \) is the confidence score of object state \( s^t_k \), returned by the object detector.

**Cross-frame Constraints:** model the spatial constraints of object states between consecutive frames. For tie plate detection, we assume the spacing between consecutive ties in the rail track is a constant. Given state \( s^t_k \) at frame \( t \), and \( s^{t+1}_l \) at frame \( t+1 \) (\( k \) and \( l \) may be different views), there are two possibilities: \( s^t_k \) and \( s^{t+1}_l \) correspond to the same physical object or, to 2 different (adjacent) physical objects.

We represent the cross-frame constraints in both those case by equation 4 as follows.

\[
\phi(s^t_k, s^{t+1}_l) = \max \left( \frac{F \left( s^t_k - s^{t+1}_l \right)}{\lambda}, \frac{\lambda}{\tau} \right) \tag{4}
\]

where \( \lambda = [\mu_f, \sigma_f, \mu_v, \Sigma_v, \tau] \), \( \langle \mu_f, \sigma_f \rangle \) models the Gaussian distribution of the object state at the next frame given its state at previous frame, \( \tau \) represents the DMI data, \( F() \) is a distance function that computes a matching score for each pair of object states \((s^t_k, s^{t+1}_l)\). Both \( \mu_f \) and \( \sigma_f \) can also be learnt from labeled training data.

### 2.2 Real-time Algorithm

We implement a real-time algorithm that can perform at the vehicle speed of 10 mph, with frame rate 20 fps. At each time point \( t \), the real-time algorithm aims at computing the optimal path from time 0 up to the current time point \( t \), given all object states from the beginning up to \( t \) as follows.

1. Compute the score for every node in the graph using dynamic programming:

\[
x^t_k = \psi(s^t_k) \tag{5}
\]

\[
x^t_l = \psi(s^t_k) \max_j (x^{t-1}_j \phi(s^t_k, s^{t-1}_j)) \tag{6}
\]

2. At each time point \( t \), select the optimal object state \( s^t_v \), where

\[
v = \arg \max_k (x^t_k) \]

3. Use the selected object states to infer/update suboptimal object states in other camera views at time \( t \).

4. If no object detection is found at time \( t \), restart the algorithm at time \( t+1 \).
**Figure 4.** Selected frames showing all 4 camera views, with the original detections (in red), corrected detections (in blue) and inserted detections (in green).

<table>
<thead>
<tr>
<th>Approach</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch Alg</td>
<td>0.84</td>
<td>0.92</td>
<td>0.88</td>
</tr>
<tr>
<td>Real-time Alg</td>
<td>0.79</td>
<td>0.92</td>
<td>0.85</td>
</tr>
<tr>
<td>Single-view detection [11]</td>
<td>0.83</td>
<td>0.84</td>
<td>0.84</td>
</tr>
</tbody>
</table>

**Table 1.** Comparative performance on tie plate detection on the rail track data with heavy occlusion and shadow issue.

4 Conclusion

We described a unified framework for rail object detection and association, based on a global optimization formulation. Our framework receives information from multiple cameras and DMI sensor, and integrates cross-object spatial constraints between objects in sequential structures, cross-frame and cross-view constraints between camera streams, in order to find the optimal set of detections over time and space. Experimental results on real rail track-driving data demonstrates that our approach achieves superior performance in very challenging scenarios. In the near future, we expect to extend our approach to handle different types of rail objects, and potentially different linear structures other than rail tracks.

References