An Automated Detection Framework for Multiple Highway Bottleneck Activations

Nguyen, T.T.; Calvert, S.C.; Vu, Hai L.; van Lint, J.W.C.

DOI
10.1109/TITS.2021.3055640

Publication date
2021

Document Version
Final published version

Published in
IEEE Transactions on Intelligent Transportation Systems

An Automated Detection Framework for Multiple Highway Bottleneck Activations

Tin T. Nguyen, Simeon C. Calvert, Hai L. Vu, and Hans van Lint

***Abstract*— Highway bottlenecks are responsible for the majority of traffic congestion. Although the problem of bottleneck detection is not new, contemporary methods have not solved the problem thoroughly with regards to bottleneck locations, activation time, and related congestion tracking. These elements are essential for identifying and characterizing a bottleneck. This paper proposes a comprehensive framework for detecting and extracting these features of highway bottlenecks from traffic data. We particularly focus on questions (i) whether a bottleneck is the primary source of congestion or (ii) whether it is activated due to congestion caused by another downstream bottleneck. The underlying principles of the proposed method include the detection of congestion (in spatio-temporal patterns of traffic congestion), and the detection of speed discontinuities in traffic data (since this is an important indicator of a bottleneck activation). The method is data-driven and automatic therefore can be easily applied to different highways and used to obtain meaningful statistics of existing bottlenecks. We have tested the method on simulated data and also demonstrated it on real data from a busy highway section in the Netherlands. The results suggest that the method is robust to different implementations, i.e. locations, of loop-detectors which measure traffic at discrete locations.**

***Index Terms*— Highway bottleneck, bottleneck detection, congestion detection, active contour, Chan-Vese, adaptive smoothing method, logistic function.**

I. Introduction

HIGHWAY bottlenecks are activated when traffic demand exceeds capacity. For example, a high inflow from an on-ramp can increase the demand on the downstream road which activates a bottleneck; or the closing of a lane reduces the road capacity which might not meet its current demand and trigger a bottleneck. These bottlenecks account for the majority of congestion that occurs on highways [1], [2]. Detecting and/or understanding the characteristics of bottlenecks, such as location, duration or delay, play a vital role in the management and control of mobility. Sensing devices like inductive loop detectors have been implemented widely to provide an essential source of information for studying traffic flow in general and bottlenecks in particular. Knowing existing bottleneck locations and their effects on traffic enables traffic experts to act quickly, albeit manually exploring or searching such data would demand considerable time and effort. To effectively utilise increasing amounts of collected data, the development of a new method that automatically analyses traffic data for bottlenecks information is necessary due to three reasons. First, by exploring traffic data automatically, such a method can simply save a lot of time and effort for network operators. Second, the automation property enables the study of bottlenecks for long periods, e.g. months or years; hence, their long-term related statistics can be obtained for further characterizing and understanding traffic bottlenecks. Finally, the method can be applied widely to large-scale freeway networks. In particular, bottlenecks on region-wide or nation-wide highway networks can be extracted automatically to study traffic network performances.

Different approaches have been proposed in the literature to identify and extract highway bottleneck characteristics automatically. Early approaches focus on pre-identified bottleneck locations, which can be learned from either network topology or historical observations. Accordingly, traffic information such as speeds or flows are obtained from related detectors (upstream and/or downstream). One can calculate the changes of speed or flow over time, and define appropriate thresholds based on historical statistics to detect the onsets and dissolves of the corresponding bottlenecks [3]–[5]. Recent research focuses on both the spatial and temporal evolution of activated bottlenecks. Instead of individual bottleneck locations, data collected from long road segments are processed for information about multiple existing bottlenecks. Speeds or flows are normally presented by spatio-temporal maps, which are essentially matrices. Chen and Rakha [6] proposed a set of image-processing techniques to classify traffic states into congested or non-congested. Thereafter, additional information is incorporated on the related network topology to eliminate discharging areas from congestion, and, inherently identify bottleneck locations that present in the original congestion. However, there are challenges from the aforementioned methods that still need to be addressed. First, contemporary approaches in the literature do not distinguish between stationary bottlenecks (at fixed locations) and temporary bottlenecks that arise when so-called wide moving jams propagate upstream. This miss-recognition could result in false alarms of bottlenecks. Second, most (if not all) of

Manuscript received January 2, 2020; revised July 15, 2020 and December 18, 2020; accepted January 15, 2021. This work is part of the research programme MiRRORS with project number 16720, which is (partly) financed by the Dutch Research Council (NWO). The Associate Editor for this article was W. Jin. *(Corresponding author: Tin T. Nguyen.)*

Tin T. Nguyen, Simeon C. Calvert, and Hans van Lint are with the Department of Transport and Planning, Delft University of Technology, 2628 CN Delft, The Netherlands (e-mail: t.t.nguyen-3@tudelft.nl; s.c.calvert@tudelft.nl; j.w.c.vanlint@tudelft.nl).

Hai L. Vu is with the Department of Civil Engineering, Monash University, Clayton, VIC 3800, Australia (e-mail: hai.vu@monash.edu).

Digital Object Identifier 10.1109/TITS.2021.3055640

the existing methods have been verified on rather simple corridors where existing active bottlenecks are away from each other, which means that they cause different (isolated) regions of congestion. Therefore, the problem of bottleneck identification simplifies to the detection of congestion. A gap remains for a method that can detect bottlenecks in more complex road corridors where multiple bottlenecks might be activated simultaneously and congestion from one of those propagates to other upstream bottlenecks. This is a relevant problem since such a method can provide more complete information about all potential bottlenecks on a road corridors (or even a network) and dependencies between bottlenecks can be investigated which is beneficial to traffic management and control.

This paper aims to develop a comprehensive framework for automatically detecting bottleneck activations in complex highway corridors. To do this, we first need to detect if there is a congestion pattern, which implies detecting its spatio-temporal extent. Afterwards, we figure out which bottlenecks contribute to the cause of this congestion. As a result, we develop a methodology (as described in Section III) that contains two main parts, namely a congestion pattern detection method and a bottleneck detection method which are described in detail in Section IV and Section V respectively. Our method relies solely on the discontinuities of traffic speeds over a certain time duration at (the small vicinity of) a location. Therefore, any type of bottlenecks, either due to road topologies or incidents, that induces congestion with decreasing traffic speeds at the upstream of bottleneck locations is of being detected. We verify this framework with simulated data in Section VI and apply in a real case study in Section VII. Besides, a relevant literature review is presented in the next section.

II. Literature Review

In this section, we review related research regarding the three main objectives, which are mentioned previously: (1) detecting the activation of a bottleneck, (2) identifying bottleneck locations and (3) tracking congestion forming upstream of a bottleneck due to its activation.

Early studies aim to identify the activation and deactivation of a specific bottleneck, of which the location is known a priori. Traffic data from e.g. inductive loop detectors are collected to provide information into traffic at the bottleneck. These data are time series showing the evolution of traffic variables like speed or flow. Banks [7] visually inspects time series of traffic speeds on both individual lanes and aggregated over lanes of the road segment associated with a bottleneck. The drops in speeds are used as the indicator of a bottleneck activation at that location. The method is formulated by defining a speed threshold (which is derived based on experiments) to filter 30-second-interval speed differentials [4]. Following Bank's approach, Hall and Agyemang-Dual [5] derive a threshold of occupancy-to-flow ratio to identify the formation and dissolve of a queue. Zhang and Levinson [3], [8] experimentally derive two thresholds of occupancy to classify traffic into 3 conditions: congested, uncongested, and intermediate. In addition, a bottleneck only be active under an extra condition that upstream is congested and downstream is uncongested for at least a minimal time, e.g. 5 minutes. Das and Levinson [9] incorporate and analyse both speed and flow information. In their method, traffic states are categorised into four phases: free, synchronised, congested and recovered. A decision diagram is introduced (based on various speed-flow conditions) to illustrate the changes of traffic states. Drops of speeds and flows are the fundamental metrics in this diagram, based on which bottleneck activation is identified accordingly. The authors also take into account upstream and downstream flows to argue if the onset of congestion at the current location is prone to the activation of bottleneck downstream. The aforementioned methods require two conditions that make them unsuitable for a comprehensive bottleneck detection method. First, bottleneck locations have to be known in advance; hence it can not locate bottlenecks but to detect activation of known bottlenecks. Second, they require parameters (thresholds) that are derived from manually analysing (local) related traffic data.

The second group of research incorporates both the temporal and spatial dimensions to identify the location and activation time of bottlenecks observed in data. The most popular method in this direction was introduced by Chen *et al.* [10], so-called Chen's method. It processes all adjacent pairs of detectors and defines a set of rules to detect if a bottleneck is activated between two locations. These rules consist of low speeds at upstream locations, higher speeds at downstream locations, and (spatially) monotonic decreases of speeds to a certain level. Upon detecting locations and activation time of all bottlenecks on a highway, a speed threshold, which is learned through analysis of traffic data, is chosen to determine if traffic is congested. Some important characteristics can be subsequently derived such as activation frequency of bottlenecks or average traffic delay. This method might work well in processing recurrent bottlenecks and extracting characteristics for future activations, which most likely require the same parameters for the detecting method. In line with Chen's method, Zhang *et al.* [11] attempted to formulate the approach in a systematic way. Specifically, the speed threshold is chosen as critical speeds, which are at the boundary between free and congested traffic, on related links. Also, a post-processing step was proposed to associate relevant activated points (indicating speed differences) together to form lines representing location and time of bottleneck activations. Bai *et al.* [12] proposed a similar approach to Chen's method, though they base their bottleneck activation definition on occupancy instead of speed. As acknowledged by the authors in the original paper [10], setting parameters for these methods require visually observing historical data, and different bottlenecks will require different sets of those. For example, Wieczorek *et al.* [13] conducted a sensitivity analysis of three parameters (in Chen's method) by testing 125 distinct sets to find the best combination. Although the Chen's method and other approaches can be effective, their parameters are sensitive to local bottlenecks; hence, extensively applying the method to different bottlenecks will not be efficient. Recently, Yang *et al.* [14] have investigated the problem from a statistical approach. The authors proposed an optimization algorithm to estimate critical speeds by fitting

the fundamental diagram (flow-speed plot) with multi-source traffic data. These critical speeds are used to detect when traffic is oversaturated. Then, the frequencies of congested states on various links can be calculated over a long period (e.g. three months in their case study). Frequent or recurrent bottlenecks are then identified by setting up a threshold on such frequencies. Hence, this approach could be suitable for identifying significantly recurrent bottlenecks instead of specific bottlenecks on a single pattern of congestion.

Activated bottlenecks cause upstream congestion, which results in slow traffic and increases travel time. Hence, to quantify the impact of a bottleneck, it is important to track the upstream congestion induced by its activation. For this purpose, traffic information like speeds is evaluated in both spatial and temporal dimensions, which constitute a 2D numerical matrix. Different methods have been proposed in the literature. As a follow-up improvement of Chen's method [10], Bertini *et al.* [15] added two more rules to mark the upstream congestion on the related spatio-temporal map of traffic state if and only if (1) downstream detector is labelled as congested, and (2) the speed at the current detector is below a speed threshold. The latter condition assures that only congested (space-time) points associated with bottlenecks are identified, which means low speeds due to disturbances and not caused by bottlenecks are ignored. Palmer *et al.* [16] suggest combining Chen's method with the FOTO model (Forecasting of Traffic Objects) and the ASDA model (Automatic tracking of moving traffic jams) introduced by Kerner [17] to improve the reliability of the resulted bottleneck detecting system. The emergence of different patterns of upstream congestion related to bottlenecks identified by Chen's method can be detected and their evolutions can be tracked by Kerner's methods. Analysing them can yield further details such as wave speed and travel time loss, which are relevant information of bottlenecks. Another direction of research classifies traffic states into two common states, namely congested and free flow. Ban *et al.* [18] use percentile speeds on multiple days of traffic data to identify regularly recurrent positions of bottlenecks. Then a speed threshold is chosen to binarize the (percentile-) speed map. Jin *et al.* [19] proposed coordinate transforming of the flow-density diagram into a different feature space (so-called Uncongested Regime Shift (URS) - Perpendicular to Uncongested regime Shift (PUS)) and used PUS as the indicator for congested traffic state. A threshold is determined experimentally. As a result, a congestion contour map of a corridor is obtained by calculating congestion frequencies at all detectors over time. These methods give a statistical view of which road stretch and how often it is affected by bottlenecks than specific location and activation time of a bottleneck. Elhenawy *et al.* [20] proposed a bottleneck identification algorithm based on the assumption that traffic states exist in two different phases, i.e., congested and free flow, and speeds in these phases follow two Gaussian distributions. Accordingly, a t-test is conducted on each space-time point to classify traffic state into one of these phases. The distribution of speeds can vary significantly between different highways; hence, their parameters need to be well adjusted before being further applied. Using the same assumption, Chen *et al.* [6] generalize the problem of classifying traffic states into two categories to image binarization, which is a well-developed topic in computer vision. The authors proposed using Otsu's method which, in essence, minimizes speed variances in individual categories as well as maximizing their variances across categories. In addition, road geometry is incorporated to separate bottlenecks which might be connected in a speed map, i.e. congested traffic from one bottleneck propagating upstream and merging to congested traffic of the downstream one. One of the main drawbacks of this approach is that the location of a bottleneck is an uncertain quantity. For example, the activation can occur at any place downstream of an on-ramp. Hence, although combining different sources of traffic data is a reasonable approach, further research is required to enable extending this method to different locations or roads.

Activations of bottleneck dampen vehicular speeds and the effect is visually strongly observable in speed maps. In addition, applications of traffic state estimators can result in satisfactory views of traffic over both the spatial and temporal dimensions with equidistant resolutions. The Adaptive Smoothing Method [21], [22] is a simple filter yet significantly effective estimator for filling in traffic information (e.g., speeds) between detector locations. The resulted speed maps are, therefore, easily seen/treated as images. Besides, computer vision is a well-developed field where tools and techniques are available for wide-range applications. Hence, image processing techniques have naturally come as a suitable approach for detecting bottleneck activations.

In summary, the literature offers a range of methods to classify traffic patterns and determine bottleneck locations and properties. However, these methods may not work on road corridors with several bottlenecks in close vicinity. In this paper, we propose a comprehensive method from an image-processing approach to (i) automatically detect locations and activation/deactivation time of highway bottlenecks, and (ii) track the congestion resulted upstream. Importantly, the method is able to disentangle different bottlenecks in complicated congestion existences in which multiple bottlenecks are being activated concurrently and causing joint platoons of traffic jam. The related upstream congestion is identified to quantify these bottlenecks' impacts on traffic. The method can be easily extended to different highways to efficiently assist (large-scale) studies of traffic bottlenecks.

III. Proposed Framework

This section presents our proposed framework for the problem of bottleneck detection and associated congestion identification, as illustrated in Fig. 1. Overall, the bottleneck detection method locates potential bottlenecks from traffic speeds. The results are validated using other relevant sources of data, for example, road geometry can be used to evaluate detected locations whether they are reasonable. The detection method, which is the core of our framework, consists of four main modules, namely congestion detection, speed discontinuities detection, activation location and time identification, and refinement.

The first module classifies speed measurements into either congested or uncongested states. Given a speed map

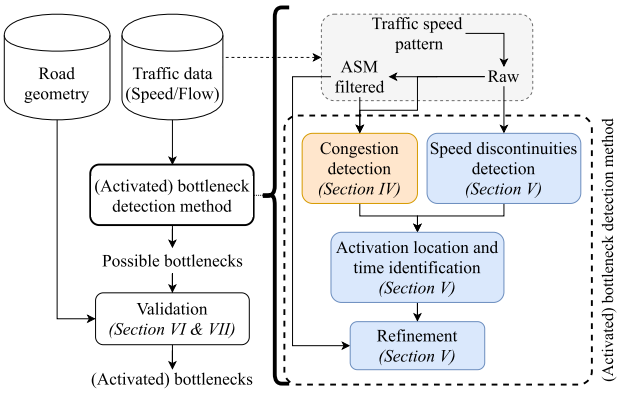


Fig. 1. The overall framework of the proposed methods for bottleneck detection.

representing traffic on a route over time, this module essentially identifies congested regions which are our main region of interest. Additionally, this also plays an important role in associating relevant (congested) regions to different bottlenecks that are going to be detected. Hence, together they can provide a more complete picture of bottlenecks including activations and consequences. The second module aims to highlight speed discontinuities in the traffic pattern since these are probably the first necessary (and easily observable) evidence for the existence of bottlenecks. This module also takes into account wide moving jams that introduce speed reductions (although they are not necessarily static bottlenecks). The embedded method can reduce their effects on detecting bottleneck-related discontinuities of traffic speeds. Based on the highlighted speed disruptions, the next module identifies potential bottleneck activations therein. To do so, it needs to gather and cluster highlighted points by incorporating their spatial and temporal information. The goal is to separate points associated with different activations of bottlenecks. Notice that, the previous modules process data from loop-detectors (so-called raw data); hence, the outcomes, i.e. activated locations, are locations of detectors. That means we obtain boundaries of potential bottlenecks but precise locations of their activations are not yet determined. This is where the filtered data, which estimate traffic data on a more fine-grain granularity, will be beneficial. We applied the Adaptive Smoothing Method (ASM) [21], [22] which is a well-known method for estimating traffic data at locations between loop detectors. The refinement method combines initial detections with ASM data to determine more precisely the locations where bottleneck congestion saturates. As a result, the locations and time of bottleneck activations are detected automatically by the various algorithms developed in these modules. The following sections will explain these modules in more detail.

In the proposed framework, raw data and ASM data are used in different modules to utilize their advantages. Raw data are direct measurements collected from loop detectors and ASM data are obtained from applying the Adaptive Smoothing Method on the raw data. While we only have traffic measurements at sparse locations where loop detectors are available, ASM further estimates traffic data at equidistant locations and provides a more complete view of traffic therein. In our framework, ASM data are used for detecting congestion because an image-based representation of a traffic pattern is more efficiently constructed from ASM data as compared to raw data; and this inherently increases the precision of the applied detection method. Since ASM is essentially a low-pass filter, it smooths out peaks explicitly in raw data. Hence, we initiate the detection from raw data and use ASM to refine results afterwards.

IV. Congestion Detection

This section presents a new approach based on image processing methods for congestion detection in the first module. Given an image representation of a congestion pattern, the objective is to detect various regions that are associated with congested traffic. We propose to formulate this as an image segmentation problem in which the target is to discern foreground objects from background areas which represent congestion and uncongested traffic respectively. We first introduce a well-known approach which is the so-called Chan-Vese model [23] from a general view on object tracking. Afterwards, we show how the model is used to formulate the congestion detection problem.

A. The Chan-Vese Model

The Chan-Vese model [23] is an active contour model which evolves a curve to boundaries of objects in images. The main principle of the algorithm is to minimize an energy function $F(c_1, c_2, C)$ defined as:

$$\begin{aligned} F(c_1, c_2, C) = \mu\ \text{Length}(C) + \nu\ \text{Area}(inside(C)) \\ + \lambda_1 \int_{inside(C)} |u(x,y) - c_1|^2 dxdy \\ + \lambda_2 \int_{outside(C)} |u(x,y) - c_2|^2 dxdy \end{aligned} \quad (1)$$

where $u(x, y)$ is a given intensity image, C is any variable curve, c_1, c_2 are average intensity values of u inside and outside C respectively, $\mu \geq 0, \nu \geq 0, \lambda_1, \lambda_2 > 0$ are fixed parameters. The solution C is at the boundaries of foreground objects in the image. For details of explanation or justification, we refer the readers to the original paper [23].

The minimization problem can be solved by using the level set method [24] which describes all computations on a level set function ϕ having the following features

$$\begin{cases} \phi(x,y) = 0, & \text{for } (x,y) \in C \\ \phi(x,y) > 0, & \text{for } (x,y) \in inside(C) \\ \phi(x,y) < 0, & \text{for } (x,y) \in outside(C) \end{cases} \quad (2)$$

The energy function is updated as a function of ϕ (see Equation 3) instead of C.

$$\begin{aligned} F(c_1, c_2, \phi) = \mu \int \delta(\phi(x,y)) |\nabla \phi(x,y)| dxdy \\ + \nu \int H(\phi(x,y)) dxdy \\ + \lambda_1 \int |u(x,y) - c_1|^2 H(\phi(x,y)) dxdy \\ + \lambda_2 \int |u(x,y) - c_2|^2 (1 - H(\phi(x,y))) dxdy \end{aligned} \quad (3)$$

where, H is the Heaviside step function and δ is the delta function; their definitions are shown in Equation 4.

$$H(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0 \end{cases}$$
$$\delta(z) = \frac{dH(z)}{dz} \tag{4}$$

Consequently, a curve C can be defined implicitly by the zero-level set of the function ϕ (i.e. set of points with $\phi(x, y) = 0$). Accordingly, the motion of a curve can be represented efficiently and easily by tracking the zero level set of the function ϕ. The minimization of $F(c_1, c_2, \phi)$ can be solved by constructing the Euler-Lagrange equation for ϕ (noting that c_1, c_2 are dependent and easily calculated from ϕ). To satisfy the differential condition, a small adjustment is made to make the Heaviside step function and the delta function differentiable at around location $z = 0$. We call these adjusted versions H_ϵ and δ_ϵ; as $\epsilon \to 0$ they converge to H and δ respectively. Now, the evolution of ϕ (over virtual time t) is described by the following Euler-Lagrange equation

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon \left[\mu \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} - \nu - \lambda_1 (u - c_1)^2 + \lambda_2 (u - c_2)^2 \right] \tag{5}$$

From Equation 5, the evolution of ϕ is controlled by two terms: the curvature $\kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}$, which preserves its smoothness and the so-called region term $-\lambda_1 (u - c_1)^2 + \lambda_2 (u - c_2)^2$ affects the motion of the (zero-level set) curve.

B. The Chan-Vese Model for Traffic Congestion Detection

In traffic congestion detection, we aim to detect the curve C that surrounds congestion regions presented in spatio-temporal speeds of a given pattern. This speed pattern is equivalent to an intensity image $u(x, y)$ where pixel values represent traffic speeds at corresponding locations x and time y. The congestion region is the $inside(C)$ and the free flow traffic region is the $outside(C)$. Based on this notation, our congestion detection problem can be solved by applying the Chan-Vese method to the equivalent image of traffic speeds. In Section VI we will elaborate the Chan-Vese method step-by-step and illustrate these steps with an example traffic data set (e.g. Fig. 4 and Fig. 5). Before doing so, we first explain the second key component of our methodology, which is the bottleneck identification method.

V. Bottleneck Identification

In this paper, we aim to detect activations of bottlenecks in two situations. Specifically we are interested in (i) whether a bottleneck is the primary source of congestion or (ii) whether it is activated due to congestion caused by another downstream bottleneck. We refer to these situations as primary and secondary bottlenecks. In the activation of the former, there is no congestion downstream of the corresponding bottleneck, meaning traffic is moving freely; whereas in the latter case, downstream of the bottleneck is congested due to another bottleneck (further downstream). In a dense network where there are multiple (topologically potential) bottlenecks located close to each other, congestion due to an activation of a bottleneck can propagate upstream and trigger other bottlenecks. Disturbances can emerge and possibly turn into wide moving jams which can pass through upstream bottlenecks. These factors might hinder the detection of activation of these secondary bottlenecks for (at least) two reasons: (1) interruptions of traffic speeds at secondary bottlenecks are normally less significant as compared to those at primary bottlenecks since traffic speeds are already low when approaching these locations, and (2) the speed changes are interfered with wide moving jam which can be observed more clearly along the direction of those jams. To avoid (falsely) recognizing the former phenomenon with any other speed disruptions (which are not due to bottlenecks), one would need to observe the disruption on a temporal dimension to test for longevity. Only if traffic has been congested for a certain long period, a bottleneck can be a possible reason (though another possibility is incidents). Regarding the second reason, it is generally accepted (i.e. there is abundant evidence [21], [25]–[27]) that the dominant congestion wave speed is in the vicinity of -20km/h (the negative sign indicates opposite direction of traffic); hence, by introducing a filter along this direction, one can expect to eliminate the interference of wide moving jam in the detection of activations of secondary bottlenecks.

Based on the above observations, we have identified and developed a method for detecting and identifying both location and activation time of bottlenecks, especially in dense networks where there are multiple bottlenecks in close vicinity.

A. Speed Discontinuities Detection

In the spatio-temporal representation of traffic, a bottleneck activation is observed by (temporally lasting) decreases of vehicular speeds at a certain location (or a vicinity thereof). This phenomena holds in bottlenecks caused by either road topologies or incidents. To identify bottleneck activation, we first detect speed discontinuities along the direction of traffic flow under congested condition. In congested traffic disturbances propagate against the direction of traffic flow. Accordingly, gradients are calculated in this direction to highlight the disruptions (if they exist) of traffic speeds. Below we develop a method to construct and apply an appropriate gradient kernel for that purpose.

Given a traffic speed pattern represented by intensity image u, Equation 6 shows the procedure of calculating gradients, G^c, along congested waves. The kernel P_c is defined by rotating a longitudinal gradient kernel P, which calculates speed differentiations on the spatial dimension. The size of the kernel determines how many related neighbouring pixels contribute to the speed discontinuity of a central pixel. Throughout the paper, traffic speeds u is presented in a way that the driving direction is from bottom to top, and that is the decreasing order of indices in u.

$$\begin{aligned} G^c &= u_0 * P_c \\ P_c &= rotate(P, w_c) \\ P &= \begin{bmatrix} +1 & +1 & +1 \\ -1 & -1 & -1 \end{bmatrix} \\ w_c &= \text{wave speed} \approx -20\text{km/h} \end{aligned} \tag{6}$$

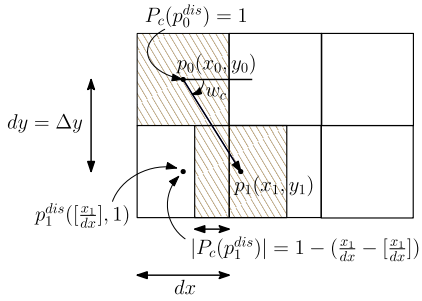


Fig. 2. A method to approximate the kernel P_c. dx, dy are temporal and spatial resolution, respectively.

One way to approximate the kernel P_c is to propagate the top row of the Prewitt kernel P to the bottom row with the speed of w_c, assuming the distance between them is Δy. This is to mimic the propagation of traffic waves in congestion. Its translated position is calculated and the corresponding values in P_c are determined by discretization afterwards. We propose a procedure as follows (see Fig. 2 for an illustration).

(i) Define a coordinate system to P with left-right and top-down as positive directions. Pick the top-left pixel and assign its coordinate as $p_0 = (x_0, y_0)$.

(ii) Translate p_0 downward with speed w_c and obtain $p_1 = (x_1, y_1)$. p_1 is identified based on the following equations. (Δy is the distance between two consecutive locations.)

$$\begin{aligned} y_1 &= y_0 + \Delta y \\ x_1 &= x_0 + \frac{y_1 - y_0}{w_c} = x_0 + \frac{\Delta y}{w_c} \end{aligned} \tag{7}$$

(iii) Now comes the discretization step with the spatial and temporal resolution (dx, dy). By assuming the distance between two rows is the unit distance, we get $dy = \Delta y$. We determine which (leftmost) item that p_1 sits on and its P_c value accordingly. This can easily be done by discretizing x_1 by the (temporal) unit dx. Accordingly, its P_c value is proportional to the intersection of cell p_1 and its discretized cell p_1^{dis}.

$$\begin{aligned} p_0^{dis} &= (0, 0), \quad p_1^{dis} = ([\frac{x_1}{dx}], 1) \\ P_c(p_0^{dis}) &= 1, \quad |P_c(p_1^{dis})| = 1 - (\frac{x_1}{dx} - [\frac{x_1}{dx}]) \end{aligned} \tag{8}$$

(iv) Fill all items on the right of p_1^{dis} with 1's and those on the left with 0's. Then construct a symmetric P_c with respect to its central item. Finally, change the sign of all values in the bottom row to negative.

This procedure can be expanded to determine kernels with more elements if needed. Note that, the above procedure uses the direct (mathematical) gradient kernel as the underlying kernel, one might as well use different kernels such as Prewitt or Sobel.

B. Activation Location and Time Identification

If a bottleneck is activated for a period, one can observe a speed disruption during that time. Alternatively, the response (G_c) of the P_c-based filter presents minimal (negative) values at related time and locations. Due to various reasons e.g. heterogeneity of traffic or disturbances of traffic at bottleneck location or noises in measurements, these negative values are not only found at the bottleneck location but also nearby locations. Besides, these locations also spread horizontally as long as the related bottleneck is activated. To aim for a robust method, we group pixels with negative values (in the response G_c) into rectangular clusters which each of them represents the speed disruption of a potential bottleneck spatially and temporally. We refer to them as *bold lines*.

A mathematical approach for this clustering problem is to determine rectangular boundaries that minimize intra-distances of pixels inside the same boundaries, pixels outside boundaries and/or maximize intra-distances between pixels inside boundaries and pixels outside boundaries. Despite that, in this section, we propose an algorithm to solve the problem with a simplified yet effective approach. The underlying principle is to find all minimum and extend the corresponding boundaries until they reach edges with average values approximately the same as the background value. The main steps of the algorithm are as follows (see Algorithm 1 for a summary):

(i) Estimate the background response value by averaging negative responses outside the congestion area.

(ii) Identify local minima in F by comparing each value with all eight of its neighbours. We denote this set as $\mathcal{M}$.

(iii) Pick the smallest minimum in $\mathcal{M}$. For each side in {left, right, top, bottom}, calculate average G^c values. If it is larger than the estimated background value G^c_{bkg}, expand the boundary to include this edge. Iterate this procedure until no more expansion is possible. As a result, a (rectangular) boundary of the region surrounding some minimal G^c can be determined. Next, remove all the minima in this region from $\mathcal{M}$ and iterate the process until $\mathcal{M}$ is empty.

(iv) Bottleneck locations and activation time: For each one of the rectangular regions found in the previous step, the location and activation time of the related (potential) bottleneck is identified by finding the line with the strongest sum/average of G^c values. A map of all these lines, indicating all possible bottlenecks in the given pattern, is obtained.

(v) Refinement: Relevant rules are applied to clean unnecessary lines, for example very short lines due to disturbances or noise. One can define the minimum activation time for bottlenecks of interest and eliminate lines with shorter lengths accordingly. Also, lines or their parts that lie outside of the congestion mask, identified in the previous section, are eliminated.

Analysis of Precise Bottleneck Locations: As previously discussed, raw traffic (speed) data preserve speed discontinuities better than ASM-filtered data. However, the detected locations of bottlenecks from the previous step are bound to detector locations which are normally sparse. In other words, the precise locations of the bottlenecks can be anywhere between upstream and downstream detectors. This section analyses potential bottleneck locations on the basis of ASM data. The ASM is based on two major assumptions: (1) free traffic

Algorithm 1 Identification of Bottleneck Activation Location and Time

Require:

Response G_c of a speed pattern to kernel P_c

Congestion (image) region indicator, or congestion mask, M_c

I - Background value estimation

Free flow mask $M_f = \overline{M_c}$

Background filter response $G^c_{bkg} = \frac{\sum_{p \in M_f} G^c(p)}{|M_f|}$

II - Local minimum

$\mathcal{M} = \{m | m \in G^c, m \text{ is a local minima}\}$

III - *Bold lines* identification

while $\mathcal{M} \neq \emptyset$ **do**

  $m_i \leftarrow \underset{m \in \mathcal{M}}{\text{argmin}}\ G^c(m)$

  $r_e \leftarrow$ the rectangular boundary of m_i

  while r_e is expanding **do**

    for each neighbour edge e of r_e **do**

      if $\frac{\sum_{p \in e} G^c(p)}{|e|} \geq G^c_b kg$ **then**

        $r_e \leftarrow r_e \cup e$

      end if

    end for

  end while

  $\mathcal{R} \leftarrow \mathcal{R} \cup r_e$

  $\mathcal{M} \leftarrow \mathcal{M} \setminus r_e$

end while

IV - Location and time identification

for each $r_e \in \mathcal{R}$ **do**

  $s \leftarrow$ vertical projection of $G^c(r_e)$

  Activation location $l \leftarrow \underset{x \in s}{\text{argmin}}\ s(x)$

  Activation time t as in r_e

end for

V - Refinement

Apply relevant constraints to eliminate unrelated lines

perturbations move in the direction of traffic flow, i.e. moving forward in the direction of traffic, and (2) congested traffic propagates upstream of traffic with a constant characteristic speed (based on experiment, Schreiter *et al.* [22] suggests a value of -18 km/h for reasonably good results).

Generally, at the downstream front of congestion, i.e. a bottleneck location in our case, vehicles accelerate according to increasing space headways. When traffic approaches the maximal speeds downstream, it will slow down the acceleration rate and gradually synchronize with downstream traffic. These observations are in line with the use of exponential smoothing kernel in ASM to smooth out traffic data between detectors (over one of the appropriate directions). Therefore, we simulate the traffic speeds profile when passing an activated bottleneck using a logistic function. The corresponding

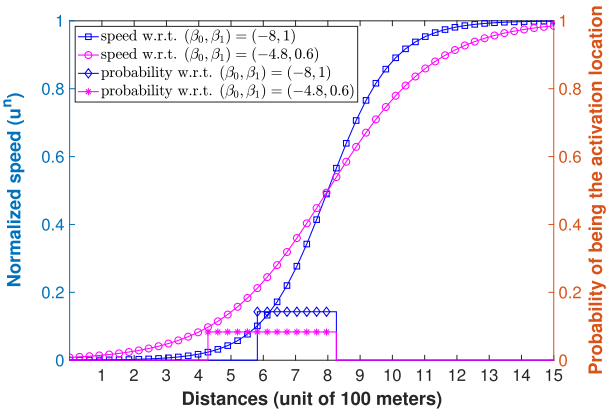


Fig. 3. Illustrations of logistic functions with two different sets of parameters.

equation and an illustrated plot are shown in Equation 9 and Fig. 3. It can be seen that the fast and slow acceleration areas are represented by the two halves of the curve. Two parameters defining the logistic function in Equation 9 are β_0 and β_1. The former, so-called *intercept* parameter, indicates spatial shifts. The latter, so-called *growth rate* parameter, represents the slope of the curve, i.e. its changing speeds from 0 to 1. This β_1 parameter, therefore, indicates how fast traffic will pick up speed at downstream of bottlenecks.

$$u^n(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{9}$$

where $u^n(x)$ represents normalized traffic speeds at location x, its boundary values, $<0, 1>$, correspond to upstream and downstream speeds of at a bottleneck respectively.

We propose two steps for identifying bottleneck activation and estimating their location more accurately. First, ASM data are used to reconstruct traffic speeds at around activated locations. Their normalized values are then fitted to a logistic curve. Second, the range of distances according to speeds from u^n_0 to u^n_1 is used to predict the possible activated location of the related bottleneck. In this work, we have chosen $u^n_0 = 0.1$ (10% of speed change), $u^n_1 = 0.5$ (50% of speed change), which is associated with the first half of the logistic functions. After the mid-point, vehicles have generally concluded their merging manoeuvres, therefore activation points should not be on this right half of the curve. We also assume a uniform distribution of the probability of precise bottleneck locations in this range. Having said that, to make a founded decision, the middle point of this range is chosen as the activation point of the related traffic bottleneck.

C. Identification of Associated Congestion

The previous sections have shown how to (1) detect congestion regions in a spatio-temporal speed map, and (2) identify lines which indicate locations and activation time potential bottlenecks therein. Based on these two elements, congestion regions associated with detected bottlenecks can be identified. There are two underlying principles in this algorithm. First, a spatio-temporal congestion region is attached to the most

downstream bottleneck (if it exists). Second, when a bottleneck is triggered, the activation continues until congestion resolves.

D. Primary or Secondary Bottleneck Determination

In this section, we propose an algorithm to determine if a bottleneck is primary or secondary. It is based on the congested regions associating to the detected bottlenecks. By identifying traffic states downstream of a bottleneck, i.e. their related congested region, the source of the corresponding congestion can be identified effectively. Particularly, the condition for a primary bottleneck is that its downstream traffic is not congested, meanwhile, congestion has already occurred downstream during the beginning of a secondary bottleneck. Our proposed procedure for identifying primary or secondary bottlenecks is shown in Algorithm 2.

Algorithm 2 Primary or Secondary Bottlenecks Classification

Require:

$\mathcal{C}$: includes a separation of related congestion regions of detected bottleneck activations

Procedure

1: **for each** $b^i \in \mathcal{B}$ **do**
2: t_0^i is when b^i is triggered/activated (which is associated with the top-left pixel of c^i)
3: $d \leftarrow$ downstream regions of c^i at time t_0^i
4: **if** d is not congested **then**
5: b^i is a primary bottleneck
6: **else**
7: b^i is a secondary bottleneck
8: **end if**
9: **end for**

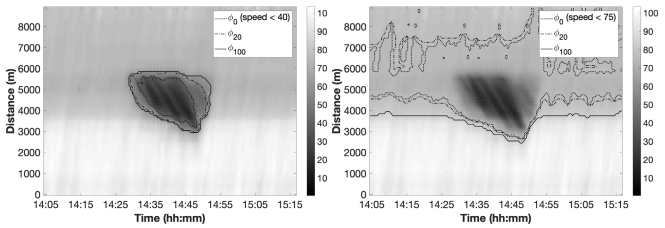
VI. Methodology Verification

In this section, we verify the two main components of the proposed method, namely traffic congestion detection and bottleneck identification. The former is compared with the bimodal-based method, a well-known method for the classification of traffic into either congested or uncongested states. For the latter, simulated data is used due to their advantages over real data.

A. Traffic Congestion Detection

To evaluate the performance of the proposed approach on classifying traffic states in congestion patterns, we first analyse the parameters in the Chan-Vese model. Then, we compare our approach with the bimodal-based method [6], which is the most popular one found in the literature.

For the Chan-Vese model to converge quickly and precisely at the boundaries of congestion regions, it is necessary to initialise the zero-level set curve, ϕ_0, close to the congestion boundaries. We have tested different initializations of ϕ_0 with various values in this range and have come to the same (expected) conclusion. Namely, using speed thresholds in between 30 and 60 km/h increases the reliability of congestion classification in traffic patterns by the Chan-Vese model.



(a) (b)

Fig. 4. Evolution of the zero-level set of ϕ according to different initializations (a) initial mask is 40km/h (b) initial mask is 75km/h.

As a demonstration, Fig. 4 shows the final contours with respect to different initial settings of ϕ_0. The presented traffic goes through two road stretches with different speed limits which impose different free speeds, congestion occurs in the downstream lower speed region and slightly reaches the higher speed region. The energy function minimization (Equation 3) has two (local) solutions on this pattern. Different initialisations of ϕ_0 lead to different classification of the pattern. If a high speed (e.g. 75km/h) is used, ϕ converges to the function whose zero-level set is at the boundary with high free speeds upstream and low speeds downstream (see the line of ϕ_0 in Fig. 4b). Consequently, the deduced congested region covers (almost) the whole region with low free speed, which is not the desired result. On the other hand, by starting with low congested speeds (e.g. 40km/h), the converged ϕ is at the boundary of the congestion that we observe from the pattern (see Fig. 4a). Hence, by starting ϕ_0 at the speed of 40km/h, the expected congested region is identified sufficiently by the Chan-Vese model.

In addition, two different scenarios have been used to compare the method with the bimodal-based method. In the first scenario, only one free traffic speed is available in congestion patterns. Note that, fluctuations of this free speed are normally observed from traffic data. The second example has at least two different free speeds in congestion patterns. Fig. 5 shows two examples of each of these scenarios and the corresponding outcomes of the two methods. As shown in the figure, both perform well on the two topmost patterns. A quantitative comparison indicates that their identified congested regions overlap by more than 99%. This shows that the proposed method based on the Chan-Vese model delivers comparable results as that of the bimodal-based method in simple layouts of road stretch, on which traffic speeds can be separated into two distributions. On the other hand, the bottom two patterns are two examples where the assumption of the bimodal does not hold. The presence of different road stretches with various speed limits has led to unexpected results when applying the bimodal method as shown in the middle figures. The congested regions are over-identified to include the lower free speed regions of the downstream road stretch. One might suggest to look for a local optimum of speed distributions in these patterns and identify the one that most likely represents expected congestion boundaries. Having said that, this depends greatly on the histogram of traffic speeds and such congestion-related optimum are not clearly shown and/or easily identified. Unlike the results from the

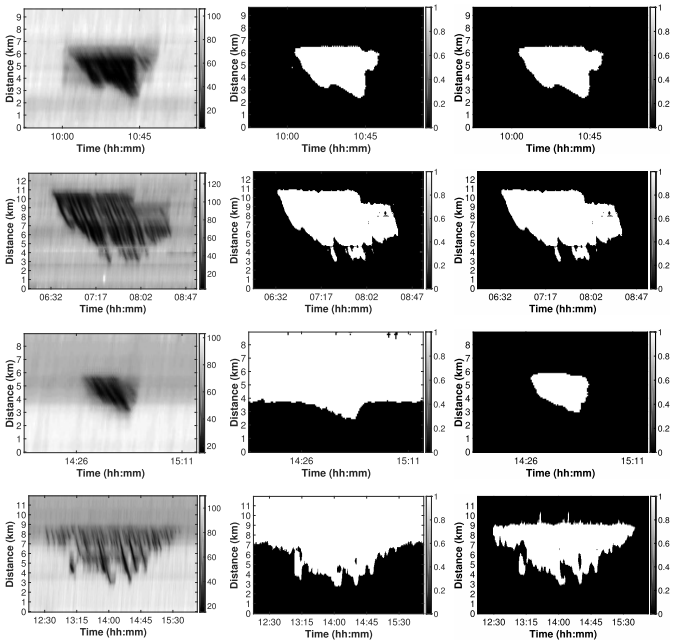


Fig. 5. Comparison of the Chan-Vese model and the Bimodal-based method on different congestion patterns: the top two patterns have one speed limit, while the bottom two have two different speed limits. The congestion detection results of applying the bimodal method and the Chan-Vese model are shown in (b) and (c) respectively.

bimodal method, those from the Chan-Vese method do not cover the entire regions with lower free speed. Qualitatively, they accurately cover the congested regions in the related patterns (as can be seen from Fig. 5c). This has shown the superiority of the Chan-Vese method over the bimodal-based method in detecting congestion in traffic patterns.

Through experimenting with the proposed approach, using the Chan-Vese model for the detection of congested regions in traffic patterns, it is positive to conclude that the Chan-Vese method is a highly viable method that can perform well on different congestion patterns.

B. Verification of the Bottleneck Identification Method

1) Verification Approach: For verification of the proposed method, we aim to analyse: (1) its capability of detecting bottleneck activations in congestion patterns, (2) how the setting of loop-detectors affects the method's outcomes. For these objectives, we make use of traffic simulation to produce crisp data that is difficult to find from real traffic flow data. In particular, a microscopic traffic simulator can provide granular details into traffic such as vehicle trajectories, traffic speeds on every short distance interval (by simply setting up loop-detectors). Note that these cannot be provided by raw traffic data due to limited numbers of loop detectors. These features enable us to identify ground-truths of bottleneck activation locations, which is necessary for evaluating the accuracy of the proposed method. Additionally, by manipulating loop-detectors (in a simulator), we can test if the method is capable of detecting activations of bottlenecks in

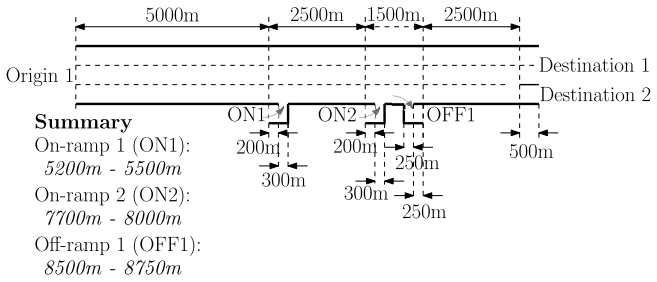


Fig. 6. The layout of the example (simulated) road stretch.

deduced traffic patterns and analyse how those settings affect the outcomes. Following are the steps carried out.

- Design a road stretch with possible close bottlenecks that are activated concurrently with a high traffic demand. This ensures that a test can be performed with heavy congestion.
- Set up loop-detectors with short intermittent distances, i.e. 100 meters, to record traffic data. This provides a convenient base for changing the loop detector setups later on. For example, we can eliminate loop detectors to get coarser traffic patterns.
- Tuning incoming traffic flows to activate one or more of these bottlenecks.
- Repeatedly apply the proposed method and investigate the results.

Details of these steps are in the next sections.

a) Simulated example design: In this study, we use the microscopic simulation tool FOSIM (Freeway Operations SIMulation) [28] which was developed at Delft University of Technology. It models traffic dynamics through the simulation of the behavior of individual vehicles. An artificial road stretch is designed as shown in Fig. 6. This road stretch has several potential bottlenecks which are two on-ramps, one off-ramp and a road split. While the first on-ramp (ON1) is located further, 2500 meters away, from the next (potential) bottleneck, the second on-ramp (ON2) is quite close to the off-ramp (OFF1) which is just 500 meters downstream. This is expected to create a complicated weaving section which will trigger congestion. The road split is designed to also create a bottleneck when vehicles have to change lanes to meet their desired destinations.

Loop detectors are implemented every 100 meters (along this 12km road stretch) and record traffic every 60 seconds. Hence, we can obtain fine simulated traffic data for further investigation. This is much better than in reality where loop detectors can be 300 meters to more than 1000 meters apart.

b) Simulated congestion patterns: Fig. 7 shows spatio-temporal speed maps of congested traffic obtained from the FOSIM model on the road layout in Fig. 6. As illustrated in the figure, two bottlenecks have been activated. The first one (ON2) is at a distance of around 8000-meter from the Origin 1, and the second one (ON1) is at a distance of around 5200-meter from the Origin 1 (see Fig. 6 for the road schematic). For simplicity, from here on we use relative distances from the Origin 1 to identify different locations on the simulated road stretch. The congestion triggered by the downstream bottleneck propagates further upstream and

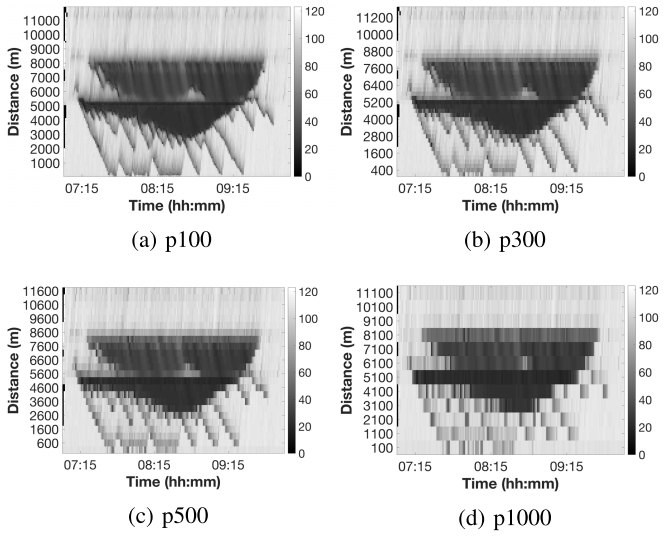


Fig. 7. Simulated traffic patterns: (a) the original pattern with detectors at every 100 meters, and other deduced patterns which are obtained by eliminating loop detectors to maintain detector spacing distances of 300m (b), 500m (c), or 1000m (d).

reaches the upstream bottleneck. Hence, we have selected this as a typical example to verify the proposed method. We have also varied distances between loop detectors to generate different levels of details in raw data. Fig. 7 shows three deduced patterns with spacings of 300, 500, and 1000 meters between two consecutive detectors. Two activated bottlenecks can be observed from these patterns clearly, although it is more difficult with those in the p1000 pattern (Fig. 7d).

2) Verification Results: The proposed method is applied to all the simulated patterns shown in Fig. 7. Summary of the results is given in Table I. It is used to answer two questions: (1) Is the method capable of detecting bottleneck activations, and (2) Do the locations extracted from different deduced patterns consistently point to the same locations? The former, once confirmed, will show the effectiveness of the method, while the latter will demonstrate its reliability.

The results indicate that the proposed method has successfully detected the activations of the two major bottlenecks in all the simulated patterns. Consequently, this simple experiment has shown the capability of the proposed method in detecting bottleneck activations or speed discontinuities in congestion patterns.

The results show that most of the associated detectors (of detected bottlenecks) are close to the activation points. In particular, the ON1 bottleneck is detected somewhere downstream of the 5200m or 5100m detectors, which are very close to the actual activated location - 5200m. Similarly, those detectors related to the ON2 bottleneck are located at 7900m, 8000m, 81000m which are also close to the activation point - 8000m. To correctly interpret these results, notice that raw data can only give rough estimates of activated bottlenecks, i.e. the locations of closest upstream and downstream detectors. Therefore, the actual locations might be anywhere in between. If intermediate locations (between detectors) are used as predicted activation points, the error that the method on raw data incurs grows as the detector spacing becomes larger (see

TABLE I

SUMMARY OF DETECTED BOTTLENECK LOCATIONS OBTAINED BY USING RAW DATA AND ASM DATA RESPECTIVELY. THESE LOCATIONS ARE SHOWN IN RANGES OF DISTANCES. FOR RAW DATA, THEY ARE UPSTREAM AND DOWNSTREAM DETECTOR LOCATIONS. FOR ASM DATA, THEY ARE ACCORDING TO 10% AND 50% CHANGES IN TRAFFIC SPEEDS. MIDDLE POINTS ARE CHOSEN AS THE FINAL DECISIONS OF BOTTLENECK LOCATIONS

		Raw		ASM	
		Detected locations	Error/ Offset	Detected locations	Error/ Offset
Bottleneck: On-ramp 1 (ON1) (5200m)					
Detector space (m)	100	5200-5300	50	5000-5200	100
	300	5200-5400	100	5100-5300	0
	500	5100-5600	150	5100-5300	0
	1000	5100-6100	400	5300-5600	250
Bottleneck: On-ramp 2 (ON2) (8000m)					
Detector space (m)	100	8000-8100	50	7700-8400	50
	300	7900-8200	50	7700-8400	50
	500	8100-8600	350	7600-8400	0
	1000	8100-9100	600	7700-8400	50

the table for details). For strong bottlenecks, like the ON1, the actual locations (5200m) are in between the detected pairs of associated detectors. Whilst this is not always the case with weak bottlenecks like the ON2, for which the detected pair (8100m-8600m) does not cover the actual activated location (8000m). There are two causes for explaining this. First, speed accelerations, i.e. magnitudes of speed discontinuities, change more sharply with stronger bottlenecks, therefore it is easier to detect their peaks. Second, since we calculate differences of speeds at locations of detectors, how those detectors are implemented also affects the accuracy of the detecting results. In particular, detection of stronger bottlenecks are more sensitive to this.

3) ASM for Bottleneck Locations: In the previous section, we provided potential road stretches of bottleneck locations from raw speeds. This section shows the application of ASM data in obtaining more precise locations of bottleneck activations. An ASM filter is configured to construct traffic speeds on the granularity of 100-meter spacing and 30-second intervals. Fig. 8 visualises fitted logistic functions to ASM speeds around bottleneck locations. The two examples in Fig. 8c, 8d show two sets of original data points and the corresponding fitted logistic curves. They demonstrate how the ASM data is well fitted to logistic functions. This also indicates the feasibility of using logistic functions to formulate traffic speeds in acceleration areas. Fig. 8a and 8b shows all the fitted curves of traffic speeds at different activated bottlenecks in all the patterns (Fig. 7). Strikingly, all the curves on the right figure, i.e. related to the ON2 bottleneck, are extremely similar (if not likely the same). Notice that, this is a weak bottleneck and traffic speeds changing slowly, therefore, requires a long distance to approach downstream traffic (free) state. The left figure, which is for the stronger bottleneck - the ON1, does not show the same result. While the differences between three fitted curves for patterns p100, p300, and p500 are relatively small as compared to 100m resolution, the curve associated with the p1000 deviates clearly from the other curves. Since traffic accelerations at downstream of a bottleneck are higher

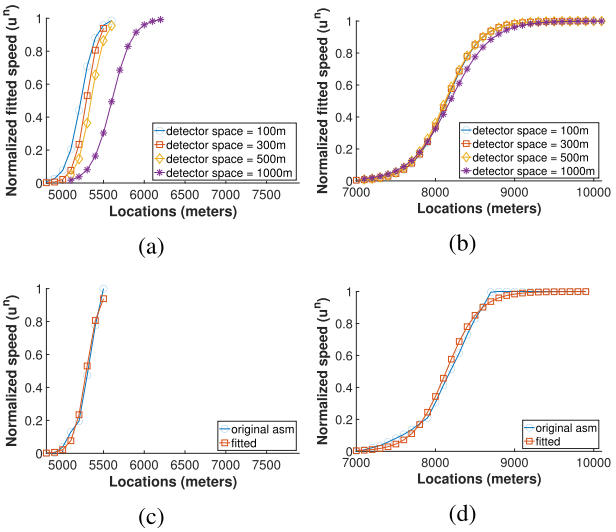


Fig. 8. Results from fitting ASM (accelerating) speeds to logistic functions under different setups of loop detectors. The two bottlenecks are (a) on-ramp 1, and (b) combination of on-ramp 2 and off-ramp 1. The scales of the x-axes in (a) and (b) are deliberately made the same to highlight the difference in the changing rates of the curves in two plots. Bottom row are two examples of fitting ASM data from 300m-apart loop detector data with respects to the two bottlenecks.

when the bottleneck is stronger, the distance that vehicles require to conclude their merging to downstream traffic is shorter. Therefore, detector spacing needs to be short enough to capture this fast change in speeds. Also, the ASM method smooths out traffic speeds in between detectors, hence the p1000 pattern subsequently underestimates the transition of speeds at downstream of the ON1. In summary, even though the ASM data can be used to estimate bottleneck locations more accurately, the improvement depends on bottleneck strengths and locations of related loop detectors.

The refinement results using ASM data are given in the right part of Table I. Apart from the finest configuration of loop detectors, i.e. 100m spacing, bottleneck locations are estimated more accurately with the application of ASM method. For the bottleneck ON2, the highest offset of 50 meters shows a high level of accuracy, especially with 1000m- detector spacing. For the stronger bottleneck, the ON1, the best results are on the p300 or p500 patterns. Therefore, it can be expected that ASM data improve the identifications of bottleneck activation locations.

From the verification using simulated data, we have come to two conclusions. First, the proposed method can detect activated bottlenecks in various implementations of loop-detectors. Notice that, the maximum tested distance is 1000m; however, as long as speed discontinuities can be observed, the method should be capable of detecting bottleneck activations. Second, traffic speed profiles can be constructed and analysed by using ASM-based filtering data to emphasize the precise locations of activations. Ideally, strong bottlenecks require more close detectors.

C. Time Complexity

The method for traffic congestion detection is based on the Chan-Vese model. The numerical solution proposed in the original paper [23] evolves the initial zero-level set over a predefined number of iterations, η. With a limited η, this method of detecting congestion has a time complexity of $\mathcal{O}(|E| \times |T|)$, where, $|E|, |T|$ are the sizes of spatial length and temporal duration of the (ASM-based filtering) speed map, respectively.

The time complexities of the three main components of the bottleneck identification method are as follow. Speed discontinuities are detected by filtering through all *pixels* in the speed map, hence, it has a complexity of $\mathcal{O}(|E| \times |T|)$. Activation location and time are identified from raw data with the complexity of $\mathcal{O}(|E^r| \times |T^r|)$, where, $|E^r|, |T^r|$ are the sizes of spatial length and temporal duration of the raw speed map, respectively. Observe that this is (greatly) dominated by $\mathcal{O}(|E| \times |T|)$ as filtered data generally has higher resolutions than raw data. Here, we potentially can ignore the complexity of the refinement of activation location using ASM data since the amount of related computation is minor.

By combining all the components complexities, it is expected that the complexity of our proposed framework is linear to the size of ASM-based filtering speed map. It is also worth to note that actual processing time also depends on the selection of parameters, e.g. the number of iterations in the Chan-Vese model.

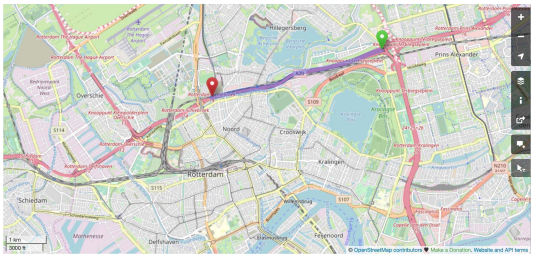
VII. Case Study

This section demonstrates an application of the proposed method. Given a route with multiple topological disruptions, like on-ramp or off-ramp, the objective is to study (1) which are the most frequently triggered bottlenecks and (2) are they the primary source of congestion, i.e primary or secondary bottlenecks. Also, we aim to have the answer over a long period, e.g. one year long, so that derived statistics can give more general overview on the road stretch. For that, an automatic method like the proposed one is highly relevant.

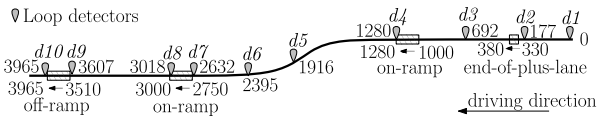
We have selected a corridor on the ring of Rotterdam, the Netherlands, to study (see Fig. 9a for a snapshot from OpenStreetMap). There are several potential bottlenecks on this stretch due to existing topological structures, namely an end of a plus lane (EoPL) – a (left) lane dedicated for fast vehicles – at around 330m-380m, an on-ramp (ON1) at around 1000m-1280m, an on-ramp (ON2) at around 2750m-3000m, and an off-ramp (OFF) at 3510m-3965m. The numbers are relative distances from the chosen route's origin which is the first detector (d1). Fig. 9b presents a simple schematic of the road stretch. Regarding data, one year (2018) of 1-minute-aggregated speeds had been collected for the whole ring road. The data are provided by the National Data Warehouse (NDW), the Netherlands [29]. We have identified 1591 traffic patterns that have congestion propagating to the selected corridor.

A. Detection of Bottlenecks on a Field-Data Pattern

One example pattern of traffic on the selected corridor is given in Fig. 10a. The objective is to detect the three activated bottlenecks. The top row shows the results with respect to the speed discontinuities detection. It can be seen that the



(a) The studied corridor on OpenStreetMap



(b) A simple schematic of the road

Fig. 9. The studied corridor is on the ring road of Rotterdam, the Netherlands. The relative distances of the detectors are shown next to the detector symbols. Estimated distances of road topologies are shown in pairs of (begin, end) distances.

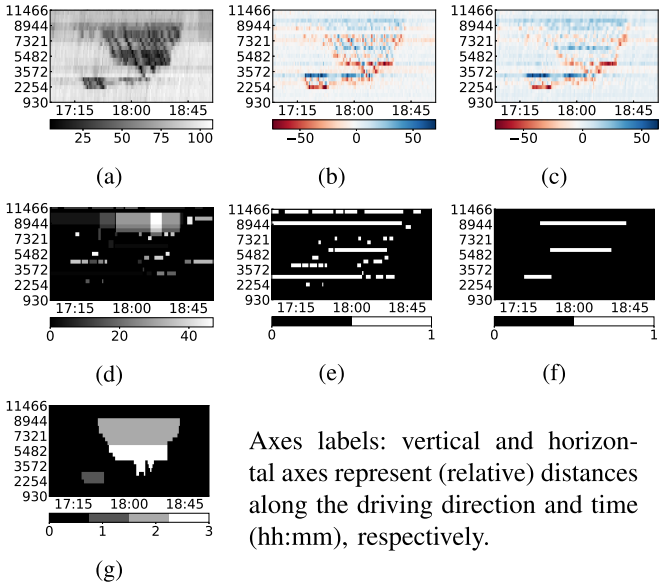


Fig. 10. Intermediate results when applying our proposed framework to a traffic pattern. Top row - a pattern of traffic speed and its responses to different kernels (with respect to the detection of speed discontinuities in Sec. V-A): (a) An example of traffic speed, (b) Response to the vertical kernel P, (c) Response to the *inclined* kernel P_c. Second row - identifying locations and activation time of (potential) bottlenecks (with respect to the proposed algorithm in Sec. V-B): (d) rectangular regions, or *bold lines*, the corresponding locations and activation time before (e) and after (f) refinement. Bottom figure: (g) tracking congestion regions associating with different bottlenecks. (For better visualisation of these plots, we refer the reader to the web version.).

response to the *inclined* kernel P_c (Fig. 10c) better highlights the locations and activation time of the three bottlenecks as compared to the response to the *vertical* kernel P (Fig. 10b). Notice that this advantage is more significant in cases that bottlenecks have high frequencies of disturbances (due to the direction on which we calculate the gradient). The second row shows the results obtaining from identifying bottleneck activation locations and time. Although different rectangles (i.e. bold

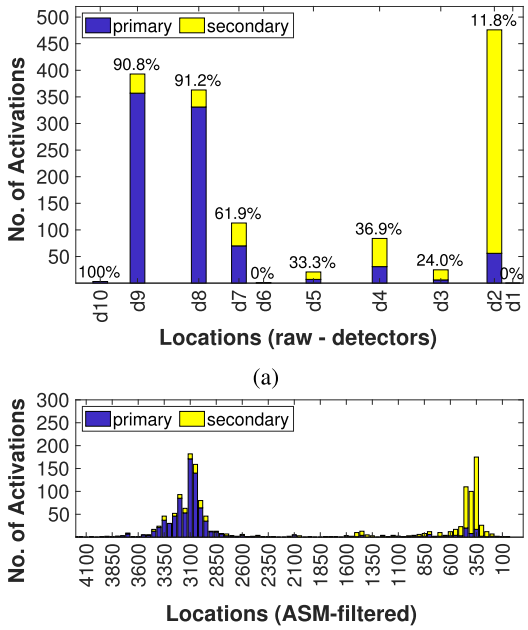


Fig. 11. Activation locations of bottlenecks on the studied corridor during 2018: (a) Bottleneck locations are associated with the most upstream detectors, with detector annotations are shown in Fig. 9b, (b) The ASM-based refinements of those raw estimations.

lines) can be detected (see Fig. 10d), their representative lines lie on the corresponding bottleneck locations. By removing the lines or parts that lie outside of the congestion region as well as those that are too short, we obtain the final result as shown in Fig. 10f. This detection result is, qualitatively, the expected outcome given the speed pattern in Fig. 10a. Also, related congestion regions are sufficiently identified for each of the detected bottlenecks as shown in Fig. 10g. This example and many others in our experiment have further confirmed the efficiency of our proposed method in detecting relevant features of activated bottlenecks from traffic data.

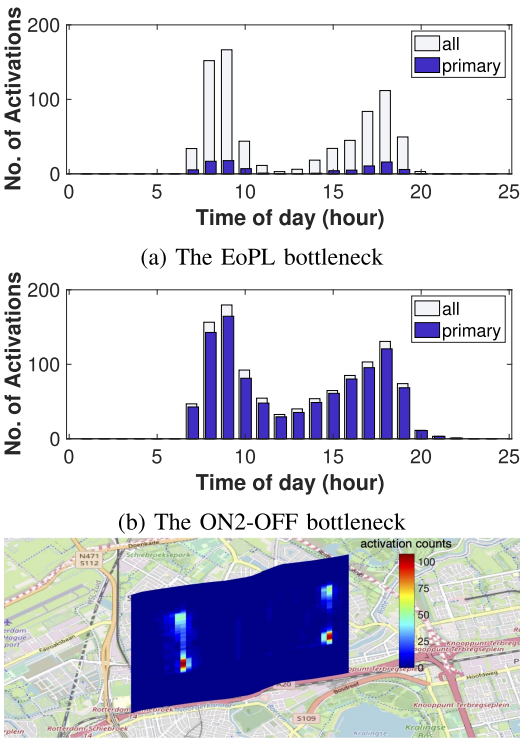
B. Derived Insights Into the Selected Corridor

Fig. 11 illustrates the outcomes of applying the proposed method to the collected data. There are several interesting findings from the left figure, which is from raw data. First, the detectors d2, d9, and d8 are the most frequently detected activation locations. Their annual counts are more than 400 times which indicate on average they are activated every day. By associating with the topology information in Fig. 9, these locations are located near three topological disruptions. Particularly, d2 is just upstream of the end of the plus lane (EoPL) on the road stretch, d8 is at the end of the ON2's shoulder lane, and d9 is at the beginning of the weaving lane before the off-ramp (OFF). Notice that, the combination of ON2 and OFF potentially creates a weaving section which causes traffic congestion. The results also provide an overview of the variance of bottleneck activated locations. While almost all the activations are determined to trigger downstream of d2 in case of the EoPL bottleneck, there are more varieties with the ON2-OFF bottleneck. This might be expected as the EoPL

is a kind of the lane drop bottleneck, and congestion usually occurs at the vicinity of the ending of the lane. Additionally, the next detectors - d1 and d3 - are quite far away from the EoPL which might explain the dominant detections of discontinuities at d2. In the case of ON2-OFF, the weaving traffic, namely trying to merge from the on-ramp and to leave the highway to the off-ramp, can create a lot of disturbances and trigger congestion when traffic is getting dense. Besides, congestion can also occur due to a high demand of ON2 merging traffic. Hence, traffic speed disruptions detected at d8 and d9 are understandable. Fig. 11a also shows much fewer amounts of potential bottleneck activations downstream of detectors d10, d6, and d1. This can be expected as there are no topologically potential bottlenecks downstream of these detectors. The detector d10 is the closest one to a physical disruption, however, it is located downstream of the off-ramp which seems to fall into the discharging areas. There are also noticeable amounts of activations at detector d4 which is just at the end of the first on-ramp's shoulder lane. The results suggest that this one does activate multiple times, although it is not as considerable as the downstream one.

Regarding the *originality* of these bottleneck activations, we have two remarkable cases to discuss here. First, the on/off-ramp bottleneck were mostly the primary bottleneck. Approximately 90% of the detected activations originate at this location. The story is opposite in the case of the end-of-plus-lane bottleneck. Nearly 90% of the occurrences, it reacts to propagations of downstream congestion. The activation intervals of all detected bottlenecks are aggregated and depicted in Fig. 12c. The plot indicates strongly the two most outstanding bottlenecks on this corridor, namely the EoPL and ON2-OFF. The heat map is also in line with the significant correlation of these two, i.e. whenever the downstream is activated, most likely the upstream will be triggered. Their specific activation time is shown in Fig. 12a and Fig. 12b, respectively. These two bottlenecks active fairly often during morning and afternoon peak-hours, although the peaks are in the morning (in both bottlenecks). The primary activation counts (over time) are also depicted to reveal if there is any correlation with activation time. In this case study, the figures suggest no indication that the chance of getting primary activations differs with respect to morning or afternoon peak hours.

For an insight into more precise locations of these bottleneck activations, the proposed method is also applied on ASM data and the results are given in Fig. 11b. First of all, there is a general trend on the road stretch, with two peaks over the corridor. Interestingly, the distributions of the activation points of the two main bottlenecks form two bell shapes. There is a difference in the widths of these shapes. The one associated with the EoPL is more concentrated in the middle, while the other one is more spread out. This is in line with the relative strengths of these bottlenecks and also with the interpretation of results from raw data. Regarding the precise locations, there are shifts in activated locations as compared to the raw data. In the case of the EoPL, the middle point is at around 375m. This is approximately at the end of the plus lane (see Fig. 9b). Although no ground truth is available (except the



(a) The EoPL bottleneck

(b) The ON2-OFF bottleneck

(c) Aggregated activation counts at all locations. The vertical axis of the heat map shows time of a day starting from the bottom.

Fig. 12. Activation time of bottlenecks on the studied corridor during 2018. Counts are aggregated over hours.

raw data which is at sparse detector locations), this finding is highly relevant to the type of EoPL bottlenecks. The ON2 was triggered mostly around the location 3100m which is just downstream of the end of the ON2's shoulder lane, although a noticeable extent is well presented. By combining with the results from the raw data, this finding seems to explain that this ON2-OFF had caused disturbances in traffic which usually saturate at the ON2 location. As a conclusion, by analysing the results with given topology, there is a confidence in ASM's capability of delivering precise bottleneck activation locations.

In conclusion, by automatically processing one year of traffic speeds, the proposed bottleneck detection method has found two most frequent bottleneck locations on the selected corridor, which is the EoPL and ON2-OFF. In addition, most of the time, the ON2-OFF bottleneck causes congestion and it, later on, triggers the EoPL. These findings suggest the majority of attention should be on the downstream location in order to mitigate the impacts of congestion and improve the quality of traffic on this corridor.

C. Time Complexity

Fig. 13 shows the realised processing time of the proposed method. The complexity of a pattern is represented by the number of measurements from related loop-detectors. It appears that there is a linear correlation between processing time and pattern sizes, which is in line with the theoretical analysis in Section VI-C. In addition, the majority of patterns

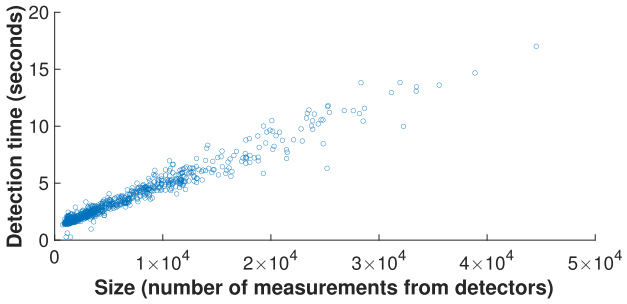


Fig. 13. Processing time for detecting bottleneck activations.

possess up to approximately 1.5×10^4 measurements and took less than 6 seconds to be processed. Hence, we can conclude that the method is efficient for offline processing bottleneck activations as well as potential for online applications.

VIII. Conclusion

This paper has presented a method to automatically detect highway bottleneck activations in congested traffic patterns using image processing techniques. First, congestion regions are identified using the Chan-Vese model, which is the active contour model without using edges. Second, the filtering kernel is constructed to detect speed discontinuities in raw traffic data, which subsequently gives approximate locations of bottlenecks. By calculating speed gradients along the direction of congestion waves, speed disruptions are efficiently highlighted. This applies to secondary bottleneck where their downstream traffic is congested; hence, assuring the detection thereof. In addition, information on the temporal dimension is incorporated to associate individual activating points at (the vicinity of) a location, which subsequently generate a comprehensive detection (represented as bold lines) of location and time of the related bottleneck. Precise activation points are inferred by fitting ASM data at those locations to logistic curves. Third, congestion associated with the detected bottleneck is identified based on the results (overall congestion regions and bottleneck activation location and time) from the first two steps. Based on that, characteristics of associated bottlenecks can be calculated such as originality of congestion, i.e. primary or secondary source. The proposed method is investigated using both simulated data and real loop-detector data, based on which we have come to the following conclusions:

- By combing both raw and ASM-filtered data, bottleneck activation locations can be determined efficiently. Raw data preserve speed discontinuities well, and ASM data support the determination of precise activated locations.
- The accuracy of detected locations (by loop-detector data, and perhaps generally fixed-location data) depends on both bottleneck strengths and locations of loop detectors. The stronger a bottleneck is, the finer detector spacing is so as to determine precisely where congestion saturates.
- Inherently from the above point, activated locations of weak bottlenecks, i.e. those with long accelerating distances, can be sufficiently determined under sparse spatial setting (500m to 1000m) of detectors.
- Logistic functions can be used to model traffic speeds at accelerating areas (downstream of bottlenecks). In addition, the growth rate parameter can be used as an indicator of bottleneck strengths.

For future studies, there are some opportunities for improving the method as following.

- The speed discontinuity kernel can be improved to not only account for congested waves but also incorporate free traffic and deceleration/acceleration area. Given that every (spatio-temporal) traffic state is classified into congested or uncongested class, this is directly feasible from the proposed method.
- A more systematic method to the identification of bottleneck activation locations from fitted logistic curves is necessary to complete the proposed method.

Since the proposed method is automatic, it can process traffic patterns and extract bottleneck-related characteristics systematically. Hence, one can easily apply the method to large-amount of highway traffic data, which is increasing quickly over time, for conveniently mining relevant information. In addition, the association of congested traffic regions to the corresponding bottlenecks provides a precise way to measure or evaluate consequences of individual bottlenecks or combinations thereof on traffic flow. In practice, the automation and advanced consequence detection bring comprehensive tools for stakeholders such as traffic manager or policymakers to get valuable insights for important tasks such as bottlenecks evaluation and strategy assessment. As a result, the impacts of highway bottlenecks can be reduced or prevented to improve mobility on highways.

References

[1] D. Hale *et al.*, "Traffic bottlenecks: Identification and solutions," United States. Federal Highway Admin., Office Oper. Res., Chevy Chase, MD, USA, Tech. Rep. FHWA-HRT-16-064, 2016.

[2] C. Systematics, "Traffic congestion and reliability: Linking solutions to problems," United States. Federal Highway Admin., Columbus, OH, USA, Tech. Rep., 2004.

[3] L. Zhang and D. Levinson, "Ramp metering and freeway bottleneck capacity," *Transp. Res. A, Policy Pract.*, vol. 44, no. 4, pp. 218–235, May 2010.

[4] J. H. Banks, "Two-capacity phenomenon at freeway bottlenecks: A basis for ramp metering," *Transp. Res. Record*, no. 1320, pp. 83–90, 1991.

[5] F. L. Hall and K. Agyemang-Duah, "Freeway capacity drop and the definition of capacity," *Transp. Res. Rec.*, no. 1320, pp. 91–98, 1991.

[6] H. Chen and H. A. Rakha, "Automatic freeway bottleneck identification and visualization using image processing techniques," in *Proc. 96th Annu. Meeting Transp. Res. Board*, 2017. [Online]. Available: https://arxiv.org/abs/1911.07395

[7] J. H. Banks, "Flow processes at a freeway bottleneck," *Transp. Res. Rec.*, no. 1287, pp. 20–28, 1990.

[8] L. Zhang and D. Levinson, "Some properties of flows at freeway bottlenecks," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1883, no. 1, pp. 122–131, Jan. 2004.

[9] S. Das and D. Levinson, "Queuing and statistical analysis of freeway bottleneck formation," *J. Transp. Eng.*, vol. 130, no. 6, pp. 787–795, Nov. 2004.

[10] C. Chen, A. Skabardonis, and P. Varaiya, "Systematic identification of freeway bottlenecks," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1867, no. 1, pp. 46–52, Jan. 2004.

[11] J.-B. Zhang, G.-H. Song, L. Yu, J.-F. Guo, and H.-Y. Lu, "Identification and characteristics analysis of bottlenecks on urban expressways based on floating car data," *J. Central South Univ.*, vol. 25, no. 8, pp. 2014–2024, Aug. 2018.

[12] Y. Bai, Z. Wu, S. Sun, and C. Wang, "Automatic identification algorithm for freeway bottleneck," in *Proc. Int. Conf. Transp., Mech., Electr. Eng. (TMEE)*, Dec. 2011, pp. 1857–1860.

[13] J. Wieczorek, R. J. Fernández-Moctezuma, and R. L. Bertini, "Techniques for validating an automatic bottleneck detection tool using archived freeway sensor data," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2160, no. 1, pp. 87–95, Jan. 2010.
[14] Y. Yang, M. Li, J. Yu, and F. He, "Expressway bottleneck pattern identification using traffic big data—The case of ring roads in Beijing, China," *J. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 54–67, 2020.
[15] R. L. Bertini and Z. Horowitz, "Diagnosing a freeway bottleneck in Portland, Oregon (USA) using archived sensor data," *Traffic Transp. Stud.*, pp. 815–827, 2008, doi: 10.1061/40995(322)76.
[16] J. Palmer, R. L. Bertini, H. Rehborn, J. Wieczorek, and R. J. Fernández-Moctezuma, "Comparing a bottleneck identification tool with the congested traffic pattern recognition system ASDA/FOTO using archived freeway data from Portland, Oregon," in *Proc. 16th World Congr. (ITS)*, 2009.
[17] B. S. Kerner, H. Rehborn, M. Aleksic, and A. Haug, "Recognition and tracking of spatial–temporal congested traffic patterns on freeways," *Transp. Res. C, Emerg. Technol.*, vol. 12, no. 5, pp. 369–400, Oct. 2004.
[18] X. Ban, L. Chu, and H. Benouar, "Bottleneck identification and calibration for corridor management planning," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1999, no. 1, pp. 40–53, Jan. 2007.
[19] P. Jin, S. Parker, J. Fang, B. Ran, and C. M. Walton, "Freeway recurrent bottleneck identification algorithms considering detector data quality issues," *J. Transp. Eng.*, vol. 138, no. 10, pp. 1205–1214, Oct. 2012.
[20] M. Elhenawy, H. A. Rakha, and H. Chen, "An automated statistically-principled bottleneck identification algorithm (ASBIA)," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 1846–1851.
[21] M. Treiber and D. Helbing, "Reconstructing the spatio-temporal traffic dynamics from stationary detector data," *Cooperat. Transp. Dyn.*, vol. 1, no. 3, pp. 1–3, 2002.
[22] T. Schreiter, H. van Lint, M. Treiber, and S. Hoogendoorn, "Two fast implementations of the adaptive smoothing method used in highway traffic state estimation," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 1202–1208.
[23] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.
[24] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, Nov. 1988.
[25] M. J. Cassidy and J. R. Windover, "Methodology for assessing dynamics of freeway traffic flow," *Transp. Res. Rec.*, no. 1484, pp. 73–79, 1995.
[26] B. S. Kerner and H. Rehborn, "Experimental features and characteristics of traffic jams," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 53, no. 2, pp. R1297–R1300, Feb. 1996.
[27] T. Schreiter, H. Van Lint, Y. Yuan, and S. Hoogendoorn, "Propagation wave speed estimation of freeway traffic with image processing tools," in *Proc. 89th Annu. Meeting Transp. Res. Board*, 2010.
[28] R. Vermijs and H. Schuurman, "Evaluating capacity of freeway weaving sections and on-ramps using the microscopic simulation model FOSIM," in *Proc. 2nd Int. Symp. Highway Capacity*, vol. 2, 1994, pp. 651–670.
[29] *National Datawarehouse of Traffic Information*. Accessed: Jan. 1, 2020. [Online]. Available: http://www.ndw.nu/en/

Tin T. Nguyen received the M.Sc. degree in computer science from the Ho Chi Minh University of Technology, Vietnam, in 2014. He is currently pursuing the Ph.D. degree with the Department of Transport and Planning, Delft University of Technology, The Netherlands. His research interest includes applying pattern recognition techniques on highway traffic congestion patterns.

Simeon C. Calvert received the M.Sc. and Ph.D. degrees in civil engineering, specialized in transport and planning from the Delft University of Technology, The Netherlands, in 2010 and 2016, respectively. He is currently an Assistant Professor of traffic and network management with the Delft University of Technology and co-leads the Delft AI Lab on urban mobility behaviour: CiTy-AI. From 2010 to 2016, he worked as a Research Scientist with TNO, Netherlands Organization for Applied Scientific Research, where his research interests include ITS, impacts of vehicle automation, traffic management, traffic flow theory, and network analysis. Much of his recent research has involved various roles in leading national and European research projects involving the application and impacts of vehicle automation and cooperation.

Hai L. Vu received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Technical University of Budapest, Hungary, in 1994 and 1999, respectively. He is currently a Professor of intelligent transport system (ITS) with the Faculty of Engineering, Monash Institute of Transport Studies, Monash University, Australia. He is a leading expert with 20 years' experience who has authored or coauthored over 180 scientific journals and conference papers in the data and transportation network modelling, V2X communications, and connected autonomous vehicles (CAVs). His research interests include modelling, performance analysis and design of complex networks, and stochastic optimization and control with applications to connected autonomous vehicles and intelligent transportation. He was a recipient of the 2012 Australian Research Council (ARC) Future Fellowship as well as the Victoria Fellowship Award for his research and leadership in ITS.

Hans van Lint received the M.Sc. degree in civil engineering informatics and the Ph.D. degree in transportation from the Delft University of Technology in 1997 and 2004, respectively. He was an Information Analyst and a Transport Engineer at various organizations. Nine years after receiving the Ph.D. degree, he was appointed as an Anthonie van Leeuwenhoek (AvL) Full Professor by the Executive Board of TU Delft in 2013 (an honour reserved for only a few young scientists and educators). He is currently the Director of the Data Analytics and Traffic Simulation Laboratory (dittlab.tudelft.nl) in which on average 15 Ph.D. students, postdocs, and programmers collaboratively work on the interface of traffic flow theory and simulation, data assimilation, and machine learning. He has authored more than 70 peer reviewed journal articles and is active in many international projects and collaborations.