Feature extraction and clustering analysis of highway congestion

Nguyen, Tin T.; Krishnakumari, Panchamy; Calvert, Simeon C.; Vu, Hai L.; van Lint, Hans

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.
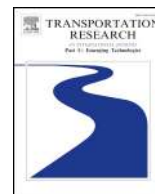
Feature extraction and clustering analysis of highway congestion

Tin T. Nguyen[a,*], Panchamy Krishnakumari[a], Simeon C. Calvert[a], Hai L. Vu[b], Hans van Lint[a]

[a] *Department of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, The Netherlands*
[b] *Department of Civil Engineering, Faculty of Engineering, Monash University, Australia*

ABSTRACT

Classification of congestion patterns is important in many areas in traffic planning and management, ranging from policy appraisal, database design, to prediction and real-time control. One of the key constraints in applying machine learning techniques for classification is the availability of sufficient data (traffic patterns) with clear and undisputed labels, e.g. traffic pattern X or Y. The challenge is that labelling traffic patterns (e.g. combinations of congested and freely flow areas over time and space) is highly subjective. In our view this means that assessment of how well algorithms label the data should also include a qualitative component that focuses on what the found patterns really mean for traffic flow operations and applications. In this study, we investigate the application of clustering analysis to obtain labels automatically from the data, where we indeed first qualitatively assess how meaningful the found labels are, and subsequently test quantitatively how well the labels separate the resulting feature space. By transforming traffic measurements (speeds) into (colored) images, two different approaches are proposed to extract the features of a large number of traffic patterns for clustering: point-based and area-based. The point-based approach is widely applied in the image processing literature, and explores local interest points in images (i.e. where large changes occur in color intensity); whereas a new area-based approach combines domain knowledge with Watershed segmentation to partition the images into different spatial-temporal segments from which domain specific features, such as wide moving jam patterns, are extracted. The results show that the Watershed segmentation separates the traffic (congestion) patterns into more meaningful and separable classes, comparable to those that have been proposed in the literature. Since there is no ground-truth set of labels, the quantitative assessment tests how well both methods are able to separate the respective feature spaces they construct for the (large) database of traffic patterns. We argue that the more crisp this separation is; the better the labelling has turned out. For this quantitative comparison we train a multinomial classifier that maps unseen patterns to the labels discovered by each of the two labeling approaches. The most important result is that the classifier using the area-based feature vector achieves the highest average levels of confidence in its decisions to classify patterns, implying a highly separable feature vector space. We argue this is good news! Not only does the combination of image processing (Watershed) and domain knowledge (traffic flow characteristics) lead to meaningful labels that can be automatically retrieved from large databases of data; this method also leads to a more efficient separation of the resulting feature space. Our next endeavor is to further refine and use this method to develop a search engine for

---

* Corresponding author.
  E-mail address: t.t.nguyen-3@tudelft.nl (T.T. Nguyen).

the (rapidly growing) 200 TB historical database of traffic data hosted by the Dutch National Datawarehouse (NDW).

1. Introduction

Highway traffic congestion is one of the central aspects concerning mobility management. To study it, traffic data has been collected for decades using various sensory systems such as inductive loops, AVI - Automatic Vehicle Identification and FCD - Floating Car Data. This includes relevant important traffic indicators like speed, flow or travel time which can be used for various purposes by road administrators, industry and academia, including policy evaluation (e.g. Van Lint et al., 2008; Zheng et al., 2012), traffic management (e.g. Soriguera and Robusté, 2011; Vlahogianni et al., 2005), traffic modelling and simulation (e.g. Wang et al., 2006; Spiliopoulou et al., 2014), to name just a few applications. National Data Warehouse (NDW), for instance, is a Dutch organisation that has been doing data collection for almost ten years over more than 7000 kilometres of freeways and provincial roads in the Netherlands. They have now stored more than 200 TB of traffic data which is a huge informative source to understand traffic dynamics.

Massive amounts of traffic data are a rich source of information; however, they also pose a challenge on how to manage them efficiently, for example regarding fast access and search-ability. Classification, i.e. adding meaningful labels to the data, is an essential step to enhance the utilisation of the storage and also to gain insights in the types of traffic congestion patterns that can be found in the collected data. A well-indexed and labelled dataset, for instance, can result in efficient search engines, which are essential for data retrieval. Both processing time and accuracy are of significant concern in this regard. The simplest way is to annotate traffic patterns manually; however, manual classification is likely only suitable for datasets with limited amounts of items since manual annotation is time-consuming and susceptible to bias. For large size datasets, an automatic approach is undoubtedly required.

Classification of highway traffic congestion has been conducted in different ways (as shown in the literature), mainly focused on either a theory-laden or data-driven approach. Theoretical approaches explain congestion by mathematical equations which describe relationships between fundamental traffic variables such as speed and flow. Hence, different taxonomies can be derived in association with supporting theories. Schönhof and Helbing (2007) simulate traffic using the nonlocal gas-kinetic based model to produce five different types of congestions - pinned localized cluster, moving localized cluster, Stop-and-go waves, oscillating congested traffic and homogeneous congested traffic. These patterns are also found in their empirical studies. Kerner (2002) defines a different taxonomy based on his three phase traffic theory. It constitutes of two fundamental types - general pattern (GP) and synchronized pattern (SP). Although these classifications are underpinned by solid conceptual and mathematical ideas about traffic dynamics, no automated methods for applying them on traffic patterns have been reported yet. In contrast, data-driven approaches instead focus on similarities between congested patterns represented in traffic data. Machine learning approaches in this direction provide more opportunities for automation. A recent idea is to conceive spatio-temporal maps as images (Nguyen et al., 2016; Krishnakumari et al., 2017). By doing so, advanced computer vision methods for image classification can be employed to classify traffic patterns. In (Nguyen et al., 2016; Krishnakumari et al., 2017), a supervised learning approach was adopted, i.e. classifiers were trained based on manually labelled datasets. These works also show that lacking of traffic-domain knowledge (Nguyen et al., 2016) or simple image processing like naive contour extraction (Krishnakumari et al., 2017), can lead to low accurate levels of classification.

Clustering analysis is unsupervised learning which aims to find an intrinsic partition of a dataset without any labelled items. Using spatio-temporal representations of highway traffic data like speed or flow, different elements of congested traffic can be noticeably observed. Thus, we aim to employ clustering analysis on highway congestion. By using images to represent traffic data, visual features can be extracted. To this end, this paper proposes and compares two inherently different feature extraction methods, namely point-based and area-based approach. While the former searches for automatic features which are motivated by computer vision, the later extracts domain knowledge-driven characteristics based on image segmentation. The case study demonstrates the ability of Watershed technique (Beucher and Meyer, 1992) in segmenting image (congestion) patterns into different segments on which corresponding features can be extracted sufficiently. In addition, by conducting cluster analysis on a dataset, a hierarchical representation of congestion with respect to their similarities is constructed. From which, typical patterns in the dataset are explored, and an initial categorization of the dataset can be created automatically and efficiently. Accordingly, appropriate labels can be generated for annotating congested patterns automatically. Finally, the effectiveness of domain knowledge in pattern representation is shown by comparing the two feature schemes.

The rest of the paper is organized as follows: Section 2 presents the literature review on the classification of congestion patterns. Section 3 represents the methodology in detail, which comprises of two approaches. Next, available data and evaluation metrics for clustering analysis are described in Section 4. Then, results and discussion are provided in Section 5. The conclusion in Section 6 justifies the work and proposes further research.

2. Literature review

There have been a significant number of studies investigating traffic dynamics with a focus on describing and understanding the resulting spatio-temporal traffic patterns. One of the main objectives is to discern different states of congested traffic. In general, there are two approaches for this, namely theory-laden and data-driven. The theoretical approach represents traffic dynamics mathematically. Related models are validated by their abilities to reproduce congested states or patterns that are observed in real life. On the

other hand, the data-driven approach explores the various observed congestion patterns and analyses their characteristics further. This section reviews some of the related works and provides discussion on the ability to automate classification. Fig. 1 provides a taxonomy of research on this topic in the literature.

A notable study in the theory-laden approach is performed by Schönhof and Helbing (2007), which employs a gas-kinetic model, which is a second-order model, to simulate traffic flow. By manipulating the relation between upstream flow (Q_{up}) and bottleneck strength, e.g. on-ramp flow (ΔQ), the model is able to reproduce five typical congested patterns, which are also confirmed by empirical data - namely pinned localized cluster (LC), moving localized cluster (MLC), Stop-and-go waves (SGW), oscillating congested traffic (OCT) and homogeneous congested traffic (HCT). Subsequently, a 2D phase diagram is constructed which correlates different congestion patterns with combinations of Q_{up} and ΔQ. This concept of analyzing these two variables is also investigated in previous research (Helbing et al., 1999; Lee et al., 2000). Although there is a solid conceptual and mathematical foundation for the method, there are two challenges for applying this phase diagram to real traffic data. Firstly, estimating (Q_{up}) and (ΔQ) is not a trivial task, and there is no accurate methodology that has been validated yet. Secondly, assuming the information on types of congestion is given, as in Schönhof and Helbing (2007), the obtained empirical phase diagram shows high interference between different congested traffic patterns. Therefore, the application of this phase diagram in the classification of traffic patterns (with real data) is not feasible.

Another well-known study is done by Kerner (2002) which is based on his proposed three phase traffic flow theory. Two main categories of congested traffic at freeway bottlenecks were introduced, namely general pattern (GP) and synchronized pattern (SP). Variations of congestion are asserted to emerge from these two main classes. According to this theory, traffic exists in one of three states, namely "free flow", "synchronized" and "wide moving jam". For recognizing and monitoring congestion patterns, Kerner developed two models - FOTO (Forecasting of Traffic Objects) and ASDA (Automatic Tracking of Moving Traffic Jams). These two models identify and keep track of traffic in "wide moving jam flow" and "synchronized flow" respectively. Consequently, every traffic state in spatio-temporal maps is automatically classified into one of the three states. There are two critical points which might limit the expansion of these models into traffic congestion classification. First, one of the underlying foundations of these models is the set of various fuzzy rules, (4 and 13 for the basic and extended sets, respectively), which consist of a number of parameters. Their values are found solely based on experiments (Kerner et al., 2004). Hence, to apply them to a different road or highway, they need to be calibrated by properly understanding the model and conducting experiments with empirical data. A systematic methodology for defining and calibrating the model still needs to be developed. Second, these models only focus on classifying traffic into three states, from which a qualitative analysis can be conducted to distinguish different spatio-temporal congested traffic, for instance, the GP or SP pattern. As for automating the classification of congestion patterns, no automatic method based on these three phases has been reported yet.

The data-driven approach constitutes two vital components - learning model and feature extraction. Depending on the availability of data, i.e. ground truth or labelled patterns, there are three learning models: supervised, unsupervised and semi-supervised (Witten et al., 2016). The supervised learning infers a pattern-label mapping function by learning from examples in a training dataset, while unsupervised learning does not require labelled patterns and explore connectivities between patterns instead. The semi-supervised learning aims to leverage the unlabelled data to improve the performance of the learners which are inferred from a few labelled items. The second component, feature extraction, concerns how to represent a pattern using specific variables. They are ideally chosen in a way to increase the chance of discriminating different patterns. Very few studies following this approach and most (if not all) of them used supervised learning method. In (Nguyen et al., 2016), a dataset of congested patterns is manually partitioned into five different classes namely isolated wide moving jam, large scale heterogeneity I, large scale heterogeneity II, homogeneous and
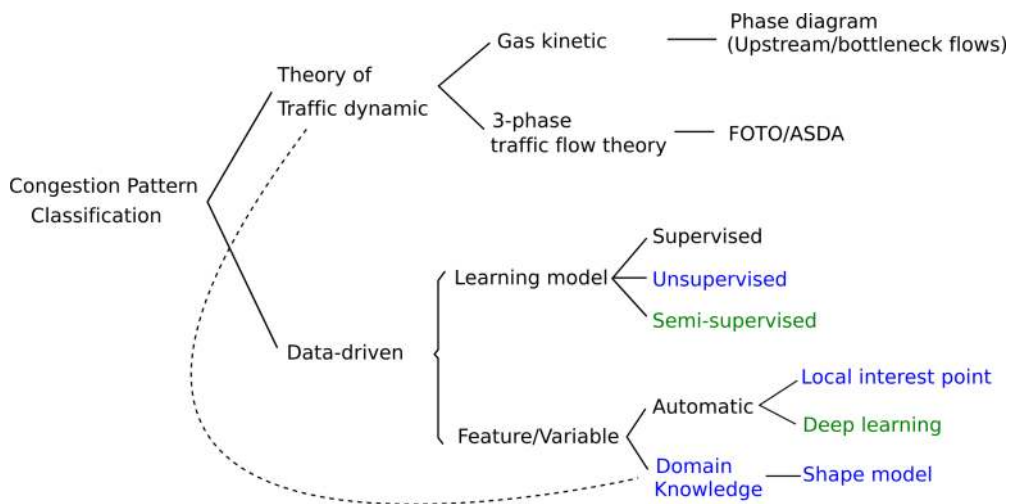

Fig. 1. Taxonomy of research on classification of congestion patterns. Color scheme code: black-existing research, green-possibility and blue-elements used in this study. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

mixed class on traffic data from two of the busiest highways in the Netherlands. This study applies automatic feature extraction in computer vision to traffic patterns. Then, a classifier based on support vector machine (Cortes and Vapnik, 1995) is trained on this dataset to classify traffic patterns. Another approach describes contours of congested patterns by shape model (Krishnakumari et al., 2017). The authors to some extent take traffic knowledge into account, i.e. using wide moving jam as one of the features. The classification accuracy is not high due to the naive contour extraction.

One of the obstacles in approaching with supervised learning is the availability of a training dataset, specifically labelled patterns. Manually processing numerous traffic congested objects is time-consuming and non-efficient as human judgment is ordinarily sub-jective and inaccurate. This can be tackled by applying methods from unsupervised or semi-supervised learning approach. In par-ticular, clustering analysis is a well-studied subject in the field of unsupervised learning which aims to divide a given dataset into different groups of items, so-called clusters. Each cluster can represent a typical pattern of the dataset. Despite the absence of a precise definition, the underlying principle is to maximize the similarity between items in the same cluster and to also maximize the dissimilarity of items from different clusters. Its applications encompass various domains and fields - data mining, text analysis, information retrieval, data annotation and pattern recognition to name a few (Xu and Wunsch, 2005). An extensive body of literature on existing clustering algorithms can be found in Xu and Tian (2015), Fahad et al. (2014), Jain (2010).

In the traffic domain, clustering analysis has been applied broadly to explore different datasets. For example, Depaire et al. (2008) cluster traffic accidents into different types given related variables such as vehicle type, gender and road type. The authors found that the employed clustering model is beneficial to the followed-up injury analysis by, for example, revealing new influenced variables for the injury outcome. Kim and Mahmassani extend a density-based clustering algorithm to cluster vehicle trajectories (Kim and Mahmassani, 2015a,b). Representative trajectories are then determined for the obtained clusters which show major traffic flow patterns over a network. Celikoglu and Silgu (2016) uses multivariate clustering, (as an extension to their previous dynamic clas-sification approach (Celikoglu, 2013)), to partition flow patterns over the fundamental diagram. The method is able to capture sudden changes or transitions of flow patterns between successive times which are promising for non-recurrent congestion detection and control. These studies have shown advantages of clustering analysis in exploring potential partitions of data, where applicable. Obtaining clusters can also assist experts in doing classification and interpreting datasets at a higher level.

This paper aims to explore the potential of unsupervised learning in traffic congestion classification. The outcomes are expected to bring first insights into a database of patterns and help annotate those automatically which certainly save a lot of manual labelling effort. We further develop and analyse the two approaches to feature extraction - automatic and domain knowledge related.

3. Methodology

3.1. Overall framework

Given a set of raw traffic speed data, the goal is to obtain an automatic classification of congested patterns. For this reason, a scheme for representing traffic patterns needs to be developed in which certain characteristics are extracted. In this paper, two different approaches are described and compared. They both conceive patterns as general images and apply different computer-vision methods to process them. One approach searches for generally local features while the other explores traffic-related characteristics. A clustering method then can be applied based on these features in an attempt to find reasonable structure of the given dataset. Fig. 2 illustrates the proposed framework including three main steps: (i) preprocessing data, (ii) extracting representative features from a pattern, and (iii) clustering the dataset based on these features. For the sake of clarity, this section briefly describes various parts of this framework.
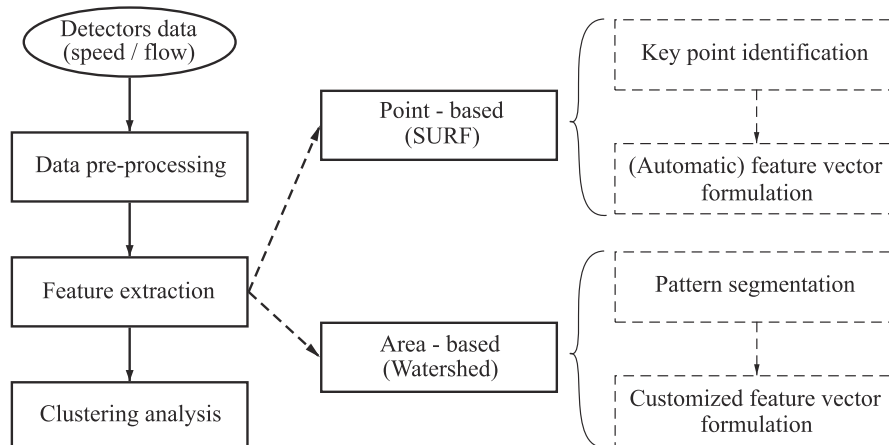

Fig. 2. General framework for pattern clustering.

3.1.1. Data preprocessing

Spatio-temporal maps have been used extensively to gain insights into traffic dynamics. Particularly, information such as speed or flow of a road stretch during a certain time period can be analyzed at a broader perspective, i.e. complete space-time view, than at just local detectors. An example of such spatio-temporal map is shown in Fig. 3a. The speed measurements are collected from the detectors distributed sparsely along the road stretch over fixed intervals.

As seen from Fig. 3a, instant speeds are represented by a corresponding point in the figure. This poses a question about traffic dynamic around those detectors where no data is available. The well-known adaptive smoothing method (ASM) serves the purpose of interpolating the unseen traffic data. For details about this technique, we refer the reader to (Treiber and Helbing, 2002; Van Lint and Hoogendoorn, 2010; Schreiter et al., 2010). The result of applying this filtering technique to the raw speed is shown in Fig. 3b. The advantage of ASM is twofold. Firstly, ASM fills in missing values and smooths out (high frequency) noise in raw speed measurements. Secondly, detectors are implemented at locations which are not necessarily at equal distances (see gaps between horizontal lines in Fig. 3a). The ASM produces an equidistant grid of smooth speeds (see Fig. 3b). Both of these points support better application of image processing techniques in the feature extraction step.

For consistency throughout the paper, we define a *pattern* of congestion as a numeric array of speeds representing (congested) traffic states over spatial and temporal dimensions. These patterns can be visualised using heat maps as shown in Fig. 3 (left: based on raw data; right: result of ASM filtered speeds). Since throughout the paper we use ASM filtered speeds, a congestion pattern in the ensuing is based on such filtered data.

3.1.2. Feature extraction

Feature extraction is one of the crucial steps to obtain an efficient representation of input patterns for data mining applications like clustering or classification. The key is to identify distinct features that make traffic patterns distinguishable from each other. The high quality of ASM filtering result (Fig. 3b) motivates a vision-based approach in which spatio-temporal patterns of congested traffic are conceived as grayscale images (or intensity images). Each pixel intensity is assigned by the corresponding speed value. These values are scaled to the range [0; 255] to fit with 8-bit representation of grayscale images. Notice that, they are, in essence, numeric matrices of filtered speed measurements. By considering them as grayscale images, we can apply advances of computer vision to traffic field.

We consider two approaches from different domains and compare their performances in clustering traffic patterns - namely point-based features and area-based features as shown in Fig. 2. As its name suggests, the point-based method explores local features which are in images. Some examples include corners, blobs or locations where pixel intensities change sharply. On the other hand, the area-based method is more about higher level features such as shapes or areas to build a customized feature vector which could potentially later incorporates traffic domain knowledge. Further details of these two methods are given in Sections 3.2 and 3.3.

3.1.3. Clustering analysis

The clustering aims at finding typical congestion patterns in a dataset. Overall, there are two approaches for clustering analysis, namely hierarchical and partitioning (Jain, 2010). We have chosen the hierarchical clustering as our main clustering method for two reasons. Firstly, this approach constructs a hierarchical representation of a given dataset which, in turn, provides an overview of the distribution of existing congestion patterns. Secondly, hierarchical clustering provides the ability of reproducibility of resulting clusters. This avoids the sensitivity to random initiations that most of partitioning clustering methods, e.g. k-means, encounter.

Hierarchical clustering can work in two different fashions - agglomerative and divisive - which are inherently bottom-up and top-down strategies for constructing a binary tree. We use agglomerative approach since it initiates each pattern as a single cluster and

(a) Raw speed   (b) Filtered speed
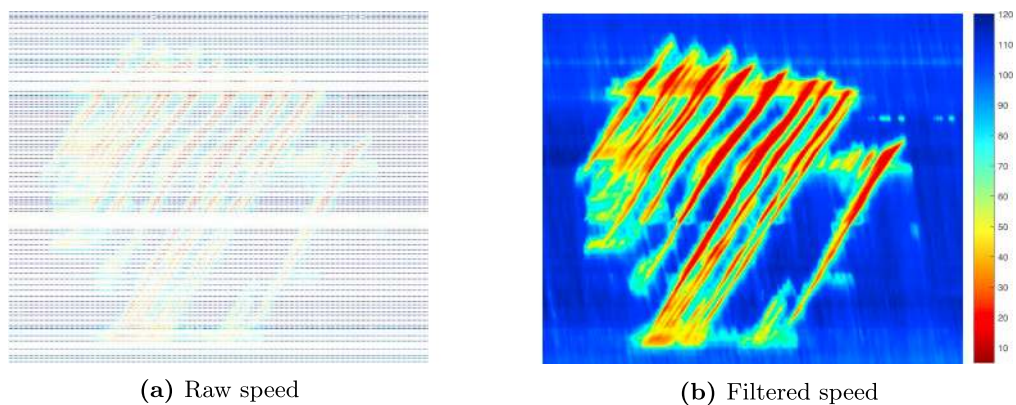
Fig. 3. Spatio-temporal speed maps of traffic on A12 highway in the Netherlands on April 12th, 2016, from 06:30am to 10:30am. The horizontal and vertical axises represent time and detector locations respectively. The driving direction is from top to bottom. For visualisation purpose, a colormap is used to emphasise different patterns inside this congestion, e.g., wide moving jams. Colors code: red implies low speeds (congested condition) while blue is for high speeds (free flow condition). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

examines connectivities between patterns or intermediate clusters. The underlying idea is to combine the two closest patterns or (intermediate) clusters into a new cluster. The connectivity between any two patterns is calculated using Manhattan distance, also known as *city block*. For the distance between two clusters, the average-link scheme is implemented which takes the average distance between all pattern pairs from those clusters (see Eq. (1)). This process continues until only one cluster remains, meaning all patterns are in the same cluster.

$$d(p, q) = \sum_{l=1}^{D} |p(l) - q(l)|$$

$$d(\mathcal{R}, \mathcal{S}) = \frac{\sum_{i}^{N_\mathcal{R}} \sum_{j}^{N_S} d\left(x_i^\mathcal{R}, x_j^S\right)}{N_\mathcal{R} \times N_\mathbf{S}}$$

(1)

where

p, q are two feature vectors representing two patterns which have D dimensions
$p(l)$ is the l_th element of vector p
$\mathcal{R}, \mathcal{S}$ are two clusters
$N_\mathcal{R}, N_S$ are the numbers of patterns in cluster $\mathcal{R}, \mathcal{S}$ respectively
$x_i^\mathcal{R}$ is the feature vector representing the i_th pattern in cluster $\mathcal{R}$

3.2. Point-based feature extraction

Using point-based features is one of the most common practices for extracting image characteristics in computer vision. The point-based features are known as interest points (or key-points) – which indicates distinctive locations in an image that contain prominent local information. As mentioned previously, the work in Nguyen et al. (2016) demonstrated the ability of this method on classifying different traffic patterns. Here we summarize this method and also refer to the original work for further details.

Generally, the point-based method constitutes two mains steps, namely (i) key point identification and (ii) feature vector formulation. It is indicated so in Fig. 2. The first step explores important points in an image, i.e. traffic pattern, which are defined by their local features. A collection of these points is expected to distinguish different images. Given detected points, the second step formulates a feature vector for pattern representation.

3.2.1. Key point identification

The key point identification step comprises of two pivotal elements - key point *detector* and key point *descriptor*. The key point detector is responsible for identifying interest points in an image. They present at various special locations such as L-corners, blobs, T-junctions or Y-junctions. These points have been widely used as local features for distinguishing between images. Several examples of such points are indicated by small circles on three different traffic patterns shown in Fig. 4. It can be easily observed that these key
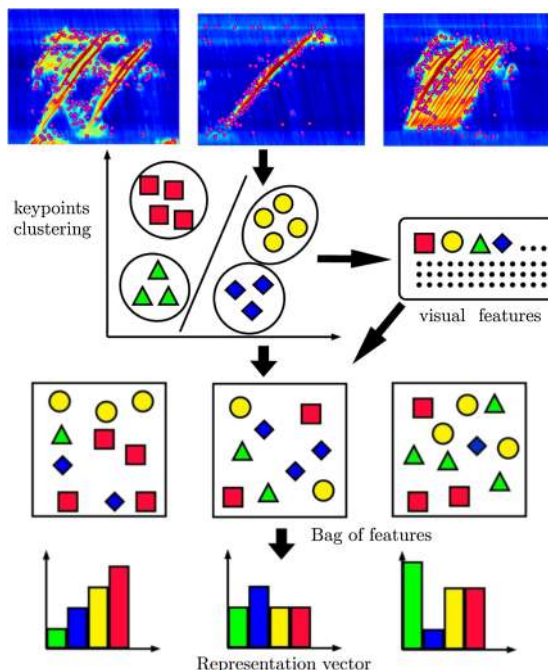

Fig. 4. Point-based feature extraction using the bag-of-visual-word method. This figure is reproduced from Fig. 5 in Nguyen et al. (2016).

points usually occur around the edges of these traffic patterns. A number of different detectors can be found in the literature (Harris and Stephens, 1988; Lowe, 1999; Kadir and Brady, 2001). Amongst those, the Fast-Hessian detector is claimed to be highly accurate and fast (Bay et al., 2008). The underlying idea is to find strong changes in surface curvature represented by intensities in an image. For that, the so-call *Hessian matrices* are constructed using second-order derivatives of the image (see Eq. (2)). $I(x, t)$ represents speed values over time and space dimensions. The determinant of the Hessian matrix is used as the indicator of key points in the input image. In addition, such determinants are also calculated in different scales of input images (so-called *scale space*) to assure the repeatability of detecting same key points in images at different sizes. A non-maximum suppression is then applied to identify maxima of determinants in 3D space (space-time-scale) as key points of the image.

$$H\left(x, t\right) = \begin{bmatrix} \partial^2 I(x, t)/\partial x^2 & \partial^2 I(x, t)/\partial x \partial t \\ \partial^2 I(x, t)/\partial x \partial t & \partial^2 I(x, t)/\partial t^2 \end{bmatrix} \tag{2}$$

The second element, key point descriptor, aims to construct a distinctive representation for key points found previously. In generally, this is done by applying an appropriate methods to represent surrounding of corresponding points. Speed-Up Robust Feature - SURF - descriptor (Bay et al., 2008) is adopted in this paper. Given a key point at (x, t, s), a window with the size of $20s$, (s indicates the scale at which the corresponding key point is found), is formed surrounding the key point. Then, the neighbourhood determined by the window is divided into 4x4 sub-regions on which Haar wavelet transform (Chui, 2016) responses are performed. This results in a 64-dimensional real valued feature vector for any given key point (the so-called SURF-64 Bay et al., 2008). We refer the readers to Bay et al. (2008) for a comprehensive description and parameter analysis related to this method.

3.2.2. Feature vector formulation

The first step results in a number of interest points in an image pattern. They are represented by a 64-dimensional feature vector. The feature vector formulation step describes a method to build representations for image patterns by accommodating these key points. It adopts the "bag-of-word" idea which is a well-known method in the topic of document classification where a document is represented by a set of its words. In that way, word order or any combinations are discarded and only their occurrence frequencies are preserved. When it comes to computer vision, it is also known as "bag-of-visual-word".

The "bag-of-visual-word" comprises of two main steps. It starts by constructing a *visual word dictionary* - an analogy with word dictionary. Since the feature vector representing key points consists of 64 real values which are continuous, they are grouped into discretised groups to facilitate the construction of such dictionary. This is illustrated in the second row of Fig. 4. In this framework, k-mean clustering method is implemented to partition all the key points collected from all image patterns. The number of clusters is specified a prior. By doing this, similar points are expected to be in the same groups and they all only represent a typical pattern of key points. Consequently, a visual word dictionary is constructed which has size as number of groups and each key in the dictionary is a typical key point, so-called *visual word*. For example, in Fig. 4, points are divided into 4 groups with different shapes/colors; hence, the corresponding dictionary comprises of 4 different visual words.

Next, all key points found in an image pattern are matched with visual words in the constructed dictionary. This is a classification problem in which we need to classify each of the detected key points to one of the visual words. A k-nearest neighbour (Cover and Hart, 1967) (knn) based classifier is trained on the clustering results from the previous step, which classifies a point into one of the visual words. Notice that, for those patterns which are used to construct the visual word dictionary, the clustering step already classify their key points into different visual words. However, for new patterns, the knn classifier is needed.

The final step is to form feature vectors for image patterns by counting the occurrences of all key points in each image pattern. Given a new image of congested traffic, interest points found by Fast-Hessian detector are classified into typical key points using the knn classifier. Next, a histogram which counts the number of points in each of the typical groups is formed and conceived as the feature vector of given pattern. The bottom part of Fig. 4 illustrates clearly these two steps. Eq. (3) formulates the point-based feature vector for a pattern of congested traffic. It can be seen that, the pre-specified number of typical key points is the dimension of the feature vector of congested patterns.

$$f_P = (n_1, n_2, ..., n_K) \tag{3}$$

where

K is the number of visual words (groups of typical key points) in the visual word dictionary
n_i is the number of i_th visual word in the pattern

3.3. Area-based feature extraction

In contrast to the point-based method, the area-based approach explores features at higher abstract levels. The authors in Krishnakumari et al. (2017), for instance, discern different congested patterns by employing the so-called Active Shape Model (Cootes et al., 1995) to represent different interested contours found in speed images. Although their approach is promising, the classification accuracy is limited due to the insufficiency of the naive contour extraction. Here, we extend this method further to overcome this shortcoming by using more sophisticated segmentation method to more accurately identify shapes of different (and relevant) elements in congestion patterns. Furthermore, a more refined feature vector is also formulated to have a more descriptive representation of congestion patterns.

We propose three steps for extracting high-level custom features from a traffic pattern: (i) segmentation of congestion patterns, (ii) feature vector formulation which extracts traffic-related features from segmented patterns and then formulate a suitable feature vector. Fig. 5 represents the outline of the process. Each of the building blocks are detailed further below.

3.3.1. Pattern segmentation

One of the main reasons for false negative classification of the classifier in Krishnakumari et al. (2017) was the contour/area detection from the traffic patterns. This segmentation step aims to divide image representation of traffic patterns into segments. Here, we use Watershed which is a sophisticated segmentation method in the field of image processing (Beucher and Meyer, 1992).

Watershed algorithm uses the gradient information of the original image. Gradients of speed images represent a space time map of (estimated) mean accelerations. The edges in speed images are detected and labelled with the maximum value of gradient. Watershed is then applied to the gradient image. This algorithm usually leads to over-segmentation and hence, a refine step - region growing - is further implemented. These steps are illustrated in Fig. 5 and detailed in the following sections.

Gradient image Gradient image describes how strong the (speed) value changes across spatio-temporal (pixel) location in the original speed map. It is calculated by Sobel method (Sobel, 1990), in which gradient is measured in both dimension of space and time. This is done by applying convolution of the speed map and appropriate Sobel operators as follows (t, x indicate temporal and spatial, respectively):

$$S_t = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}, \quad S_x = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The horizontal and vertical gradients (G_t, G_x) of the pattern, which has speed matrix indicated by $I(x, t)$, are calculated as shown in Eq. (4) (in which, ⊗ indicates convolution operation).

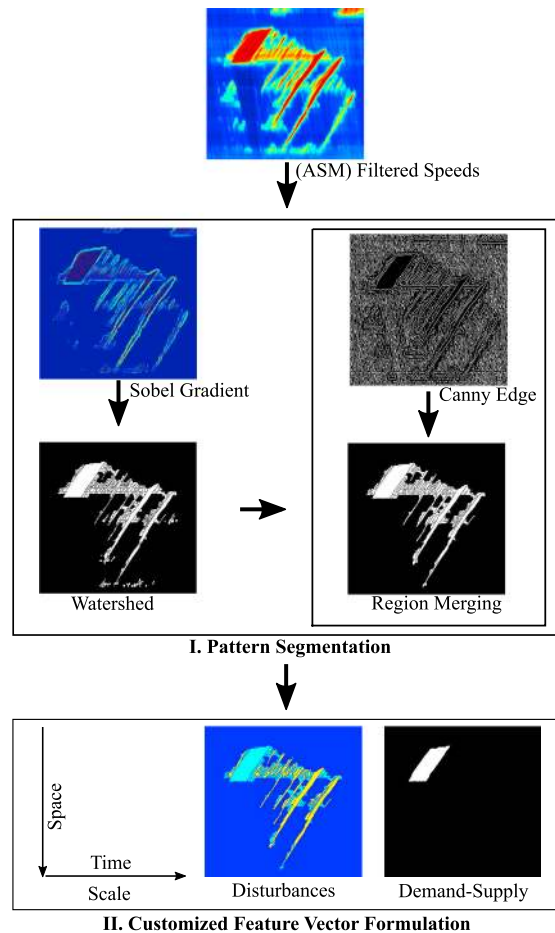$$G_t = S_t \otimes I$$
$$G_x = S_x \otimes I \tag{4}$$

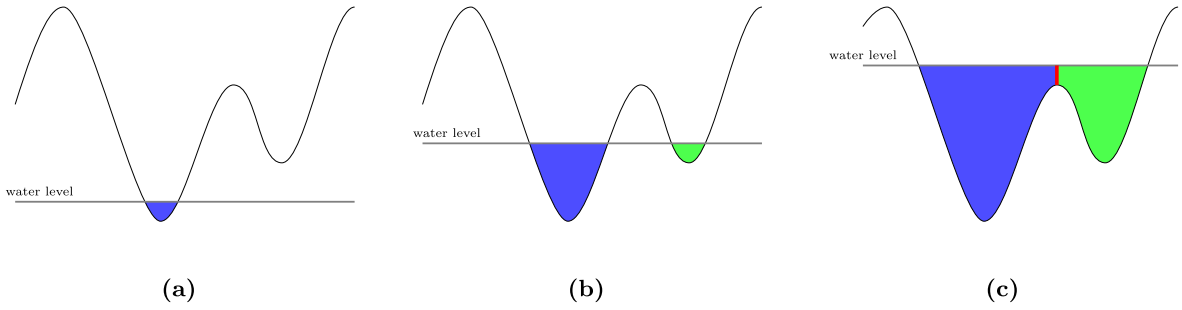**Fig. 5.** Area-based feature extraction using the segmentation approach.

Fig. 6. Demonstration of fundamental steps in Watershed segmentation. Water level reaches the first minimum (a), water level reaches the second minimum (b) and red dam prevents water merging (c).

Eq. (5) combines these multicomponent gradients to yield the magnitude of the final gradient.

$$G = \sqrt{G_t^2 + G_x^2} \tag{5}$$

The last step is to normalize the gradient value.

$$G_0 = \frac{G}{\max(G)} \tag{6}$$

Watershed Watershed is one of the main morphological operators in the field of mathematical topology which has been applied in image segmentation (Beucher and Meyer, 1992). The basic concept of this method is to consider a gray-scale image as a topographic surface in which altitude of a point is set to its intensity value. Fig. 6 demonstrates the concept using a simple 2D signal. The surface is gradually immersed into water. Virtual holes at all minima can let the water rise through. When a minimum is reached by the water, a (catchment) basin is formed accordingly. As shown in Fig. 6a, water level will first reach the lower minimum, and then water will spread into the surrounding area. A blue basin is formed for this minimum. A moment later, water will then reach the second minimum and also form the green basin (see Fig. 6b). Water level continue rising and when it reaches the local maximum, the water from two basins are about to meet. At this point, a dam, which is labelled by red colour, is built to prevent this as depicted in Fig. 6c. Dams will get higher as water level gets higher. This process continues until the surface is completely flooded. As a result, we will get various image segments, in different colours, separated by shed lines or dam as simply shown in the figure.

Algorithm 1 presents the Hill climbing algorithm for Watershed segmentation (Rambabu and Chakrabarti, 2008). It illustrates precisely the principle of immersion/flooding process above, although it is derived from topological-based definition of Watershed transform. $\overline{S}$ comprises the top part of the topographic surface that is above the water level and 1-pixel high part that is just reached by the water (In initial state, only the boundary of this part is included). The pixel x taken from step six has the lower intensity, i.e. the pixel that is just reached by the water level. The algorithm assumes water growing from this x to its neighbors x'. If x' has not been labeled yet, it will be assigned to the same basin with part of water that touched it, i.e. x's label. On the other hand, if it is assigned to a different basin, water from two different basins meets here and hence, this should belong to the watershed line. The algorithm continues until all pixels are visited.

Algorithm 1. Hill Climbing algorithm for Watershed transform

Require:
    Gray scale image represented by intensity function f
    Set of regional minima $\mathcal{M} = \{m_i\}$, $i = 1: M$ with value α_i respectively
Initialization
1: **for each** $m_i \in \mathcal{M}$ **do**
2:     $\forall\, p \in m_i$, $label(p) \leftarrow i$
3:     $S \leftarrow \{p \in m_i \,|\, \forall\, p' \in N_{G(p)}: f(p') = \alpha_i\}$ /∗ All interior points of regional minima ∗/
4: **end for**
5: $\overline{S} \leftarrow \{p \notin S\}$
Processing
6: **while** $\overline{S} \neq \varnothing$ **do**
7:     $p = \underset{p \in \overline{S}}{\arg\min} f(p)$ /∗ Take the lowest point ∗/
8:     $\overline{S} \leftarrow \overline{S} \backslash p$ /∗ Remove p from set $\overline{S}$ ∗/
9:     **for each** $p' \in N_{G(p)} \cap \overline{S}$ **do**   /∗ neighbors of p in $\overline{S}$ ∗/
10:       **if** $label(p') = Unknown$ **then**
11:         $label(p') \leftarrow label(p)$
12:       **else if** $label(p') \neq label(p)$ **do**
13:         $label(p') \leftarrow SHEDLINE$

```
14:        end if
15:      end for
16: end while
```

Region merging This step is to deal with the common over-segmentation consequence of watershed. The strategy is to merge a region with its neighbors, if reasonable, to obtain a larger region. The criteria to determine where to merge two neighbor regions is that their speed difference should not be too large. Consequently, edges should not be present between these two regions. Therefore, canny edge (Canny, 1986), an extremely well-known technique in image processing, is used as a criteria for region merging. In principle, it comprises of four main steps: (1) image is smoothed out by a Gaussian filter in order to remove noise, (2) Sobel gradient is calculated on the smoothed image, (3) applying non-maximum suppression to find local maximum intensities in gradient image which correspond to edge points, and (4) two upper-bound and lower-bound thresholds are used to filter out unwanted edge points.

The proposed algorithm for this merging is represented in pseudo-code, see Algorithm 2. Given a region, the general idea is to repeat merging it with its appropriate neighbors. The common boundary can be extracted by using fundamental morphological operators such as dilation and erosion (Haralick et al., 1987). The former operator dilates a connected region in binary images with respect to given number of pixels, while the latter one does exactly the opposite. The dilation operator expands a region to surpass the shed line separating neighbor regions. Hence, the common boundary can be identified by the intersection of these expansion parts. Afterward, erosion operator shrinks the obtained boundary to its original size for further steps.

Algorithm 2. Region merging algorithm

```
Require: R: set of regions, E: set of edges
 1: for each r ∈ R do
 2:   repeat
 3:       N_r ← Neighbours(r)
 4:       isNewMerge = false
 5:       for each n_r ∈ N_r do
 6:           c ← common boundary of n_r and r
 7:           if c ∩ E = ∅
 8:               merge n_r and r: r ← r ∪ n_r
 9:               isNewMerge = true
10:           end if
11:       end for
12:   until ¬isNewMerge
13: end for
```

3.3.2. Customized Feature Vector Formulation

To have a better view on which features we should extract from a pattern of congestion, we look further into the pattern in Fig. 5. This pattern is significantly notable in the sense that it encompasses different typical elements. A severe congestion occurred which is likely caused by an accident and results in homogeneously and heavily congested traffic - represented by a large region of red colour in the related speed image. For the sake of simplicity regarding name convention, we refer to this as (heavily congested) *demand-supply component*. Following up are numerous disturbances which spill back against driving direction. Being motivated by this, a two-level hierarchical definition of congestion pattern is formed, namely phenomenon and pattern levels. At the lower level, three important elements related to traffic flow domain are explored to describe a congestion pattern: space-time scale, disturbances, and demand-supply components. The first element approximates the extension of a congestion pattern by measuring its temporal and spatial extent. The second element is identified in Krishnakumari et al. (2017) as the most common traffic phenomenon where both of its upstream and downstream heads move against traffic direction. The last element has downstream front be stationary for longer time spans and heavily congested traffic upstream. At the higher level, a congestion pattern can be conceived as a combination of lower-level elements. The area-based approach aims to identify this higher-level congestion pattern. Following sections describe those lower-level elements in further details.

Congestion scale The congestion scale encompasses two measurements: spatial and temporal extents of congestion. The former is the total length of the road stretch being reached by traffic congestion. The latter indicates approximately how long the congestion lasts. For each location, the duration for which congestion has occurred is calculated. The highest value (over space) is used as the representative duration for the temporal extent. An illustration is shown in Fig. 5.

Disturbances identification Disturbances are observed in traffic very often. They occur either as a small disturbance inside synchronised traffic near a bottleneck or as a wide moving jam. Krishnakumari et al. (2017) successfully proposed using Active Shape Model to identify WMJs. Since small disturbances and wide moving jams have similar shapes except for their sizes, this paper further employs this method to determine disturbances in congestion patterns.

The Active Shape Model technique (Cootes et al., 1995) describes a shape using a mean shape and its variations derived from a set of similar shapes. Thus, given a new shape, the error while fitting the shape model to this shape can be used for identifying/ classifying the shape. Krishnakumari et al. (2017) gives a detailed explanation of how to build a shape model of WMJ and how it can

be used as a shape classifier. For this work, we only consider the WMJ shape, so the feature vector for classifying a given shape slightly varies to include (i) fitting error to WMJ shape model, (ii) affine transformation parameters of the fitted shape, (iii) time extent of WMJ shape, and (iv) space extent of WMJ shape. A simple logistic classifier is used on this feature vector to identify whether a shape is a WMJ (meaning disturbances in our case).

Demand-Supply identification In contrast to disturbances, demand-supply (DS) related elements commonly have downstream head being fixed at a location. In this work, we defined noticeable DS elements with following criteria: (1) average speed is equal or lower than 30 km/h, and (2) downstream head is stationary for at least 15 min. This time period is chosen arbitrarily and is certainly changed according to application interest. In general, the longer this time is, the longer congestion is experienced at the bottleneck; hence severer the situation is. They are formulated as in Eq. (7). Due to the existence of noise in traffic data, the second requirement is relaxed as having congested traffic for at least 15 min at downstream part of the region. This means the location of long lasting congestion could be at any location downstream of a congested region. We have taken 30 percent of corresponding (congested) roach stretch as the searching area for this location.

$$isDS(r) = \begin{cases} 1 & \mu_{speed(r)} \leqslant 30 \text{ km/h and congested\_time(downstream head)} \geqslant 15 \text{ minutes} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where

r is a congested region
μ denotes average

Feature Vector Formulation Based on these three feature components, we construct a feature vector for representing a traffic pattern as follows:

$$f_A = (l_{Space}, l_{Time}, n_{Disturbances}, isDS) \quad (8)$$

where

l_{Space} is the spatial extent of the congestion
l_{Time} is the temporal extent of the congestion
$n_{Disturbances}$ is the number of disturbance instances
$isDS$ indicates if any Demand-Supply elements exist in the congestion

In order to equalise the degree of influence of each feature dimension on the clustering decision, we normalised both the feature vectors from the two feature schemes f_P and f_A to the range of (0,1).

3.4. Synthesis

This section has described in depth three fundamental components which constitutes the whole procedure required to cluster a given dataset of congestions. The preprocessing component uses ASM to turn traffic data in spatio-temporal dimension into corresponding image representation which allows computer vision techniques be applied upon. Particularly, two feature extraction approaches, i.e. point-based and area-based, explore various types of features which are special local points or domain knowledge characteristics. Watershed segmentation is proposed as a method for segmenting a congestion (image) pattern into different areas. Finally hierarchical clustering will be used to analysis structure of the given set of congestions. In the following sections, the proposed framework will be applied to a case study and their performance are assessed and compared.

4. Evaluation methodology

The methodology for evaluating the different elements of the framework is designed and presented in this section. Firstly, the dataset used for applying the proposed clustering methodology is introduced. Then, comprehensive methods are described to evaluate the two main elements of the proposed framework - (i) Watershed segmentation performance, and (ii) clustering results from the two feature schemes. Watershed segmentation will be evaluated for its ability to divide a congested pattern into different spatio-temporal areas so that the domain-knowledge characteristics are derived. Similarity of intra-class patterns and dissimilarity of inter-class patterns are interpreted to evaluate the clustering results.

4.1. Dataset

In this study, we use speed data from the loop detectors as input for testing the two approaches. This data is provided by National Data Warehouse (NDW), which is an organization managing a large amount of traffic data from most of the Dutch highways. We chose two busy roadways - A12 and A13 (A12 highway; A13 highway) - for our case study. The data is collected for four months: from April to July in 2016 (This time period is chosen arbitrarily). Congestion patterns are searched on every single day via the following steps. First, smoothing method - the Adaptive Smoothing Method - is applied to obtain an image representation (a speed matrix) of

traffic state during that day. Parameters of the filter are set based on the settings in Schreiter et al. (2010); some basic ones are shown in Table 1. Second, a speed threshold of 80 km/h is used to filter congested pixels in the image. Then, relevant image morphological operators such as opening or dilating are applied to remove noises and connect nearby congestion related pixels, which have speed values lower than the speed threshold. This results in connected spatio-temporal regions of congested traffic. Bounding boxes of these connected regions are generated to extract congested patterns thereafter. Applying this for the A12 raw dataset results in 232 congested traffic patterns which constitute a primary dataset for our analysis. The two proposed methods will be evaluated on this A12 dataset. In addition, we obtained 196 patterns on A13 which is used as a secondary dataset for further evaluation as shown in Section 5.3.1 - Qualitative. Table 1 summaries some details related to the parameters used in this study.

4.2. Watershed segmentation evaluation

Watershed segmentation is the core component in the area-based feature scheme; it divides a congestion pattern into different spatio-temporal areas, particularly to separate disturbances. Since having a ground truth set is not trivial, we visually assess the segmented patterns. An important qualitative criterion is a clear separation of disturbances in traffic congestions. Various examples with multiple levels of complexity are chosen to evaluate the Watershed segmentation results. In addition, since one of the main reasons for adopting this technique is to overcome the shortcoming of previous contour extraction, a qualitative comparison between these two techniques is conducted.

4.3. Clustering evaluation

Since cluster analysis is an unsupervised learning problem, an appropriate validation is not trivial. However, there are generally two validation criteria in the literature: internal and external (Rendón et al., 2011; Jain, 2010). An internal criterion validates a clustering result based on properties inherent to the given dataset. In contrast, an external criterion matches a clustering result with prior information on structure of the dataset, which is generally referred to as *true labels*. Such information is, however, normally either subjective or unavailable. Therefore, in this work, we focus on internal criteria to evaluate the clustering results. Our evaluation metrics comprises of qualitative and quantitative measures. The former examines patterns from each of the resulting clusters by visually assessing their similarity. The latter evaluates how well clusters are separated from each other by using quantifying the confidence level of appropriate classifiers trained on the corresponding clustered data.

4.3.1. Qualitative evaluation

The qualitative evaluation step assesses the *similarity* of patterns in each of resulting clusters. There is no quantitative definition of similarity, instead we base our assessment on the appearances of congestion patterns over space and time, equivalently to compare images of these patterns. Attempts to draw some common characteristics on these patterns are driven by traffic knowledge, which has received significant attention in the literature such as (temporal, spatial) scale of congestion or disturbances.

4.3.2. Quantitative evaluation

For quantitative evaluation, we analyse the *separability* of resulting clusters. This is motivated by the follow-up application in which new traffic patterns are classified preferably without repeating the clustering for the whole dataset (previous and new patterns). Hence, resulting clusters is used as a training set, on which a suitable classifier is trained to classify new patterns. The separability criterion is then measured by the *confidence level* of this classifier's decisions to assign patterns to clusters. It is generally expected that well separated clusters would confuse the classifier less, meaning higher confidence in classification.

To quantify the confidence level of a classifying decision, we adopt the family of classifiers which give the membership probability that a pattern belongs to a class. The higher the probability, the stronger the classifier believes in its decision. Since the number of clusters can be higher than 2, we have a multi-class classification problem. In this work, we choose multinomial logistic classifier as our base classifier for evaluating the clustering results. A brief description of this method is provided later in this section.

Algorithm 3 summarizes the procedure for quantitative evaluation of clustering results. K-fold cross validation is used to obtain a

Table 1
Parameters and a few descriptions of the traffic patterns used in this study (A12 and A13 highway, the Netherlands, from April to July, 2016).

Adaptive Smoothing Method	Temporal resolution	0.5 min
	Spatial resolution	100 m
	Traffic wave speed	-18 km/h
Congested patterns extraction	Congested speed threshold	80 km/h
A12 dataset (primary)	Number of patterns	232
	Time range	10–320 min
	Space range	2.4–40.4 km
A13 dataset (secondary)	Number of patterns	196
	Time range	10–248 min
	Space range	2.4–18.8 km

stable result. It starts by dividing the whole dataset into a number of groups (step 1). In each run, one fold is taken as a test set and the rest are combined into a training set. We perform cluster analysis on this training set to obtain clusters and subsequently class labels for its patterns (step 2). Next, a multinomial logistic classifier ($\mathcal{F}$) is trained on these samples (step 3) and then applied on the test set (step 4). $\mathcal{F}$ provides membership probabilities of a given pattern to all classes. The class associated with the highest probability is chosen to label the pattern This probability also shows the confidence level of the classifier. We apply $\mathcal{F}$ to all patterns in the test set and collect all confidence levels (step 5) to evaluate the separability of the clustering results. In particular, The distribution of these probabilities are subsequently constructed for different number of clusters. We compare these for the two feature schemes to achieve a quantitative evaluation of clustering results.

Algorithm 3. Quantitative evaluation scheme of a clustering result

---

Initialization
1: Randomly divide the dataset into N folds (groups): $\mathcal{D}_i$, $i = 1: N$
Evaluation
2: **for each** fold $\mathcal{D}_i$ **do**
3:     Cluster the set of other folds $\{\mathcal{D}_{j, j \neq i}\}$
4:     Learn a multinomial logistic classifier, $\mathcal{F}_i$, given the obtained clusters as a training set
5:     Fit $\mathcal{F}_i$ on every pattern in $\mathcal{D}_i$. Collect classifier's confidence levels which are the probabilities that $\mathcal{F}_i$ provides to support its decisions to classify patterns (to one of the obtained clusters)
6: **end for**

Multinomial logistic regression The multinomial logistic regression (MLR) (Böhning, 1992) is an extension of the well-known binomial logistic regression that deals with multiclass classification problems. A softmax function is used instead of a logit function to calculate the membership probability that an observation (or pattern) belongs to one particular group. Thus, it is also called softmax regression. The predicted probability of ith category is calculated by Eq. (9), given a feature vector x and a coefficient matrix W. The input for the softmax function is a linear combination of feature vector and coefficients of class i, which is the corresponding row in W.

(a)   (b)   (c)   (d)
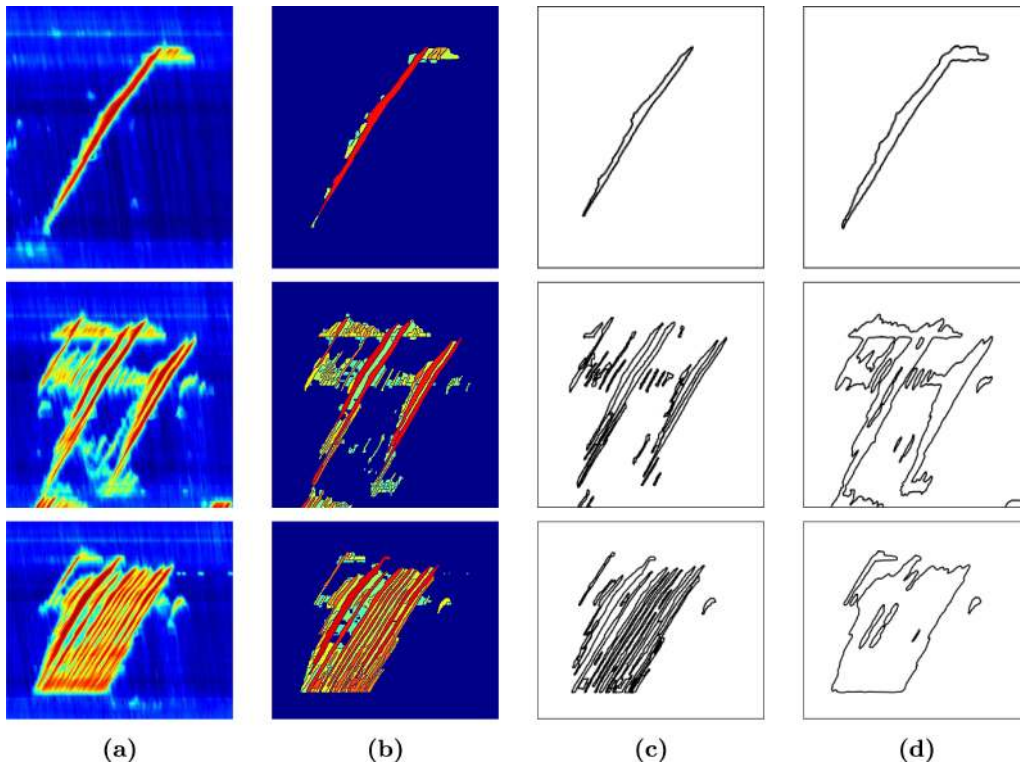
Fig. 7. Examples of segmenting congested patterns into different areas. The left column (a) are some various congestions in spatio-temporal representations. Results of the Watershed segmentation is shown in column (b). The corresponding contours extracted from the Watershed-based segments and the naive contour extraction are in column (c) and (d) respectively.

$$P\left(y = C_i \bigg| x, W\right) = \frac{e^{W_{C_i}^T x}}{\sum_{j=1}^{C} e^{W_{C_j}^T x}}$$

(9)

The coefficient matrix W can be optimized using different methods to fit the MLR model (Huang et al., 2010; Yu et al., 2011; Gao et al., 2007). In this work, we use the dual coordinate descent method for the optimization (Yu et al., 2011).

5. Results and discussion

This section presents and discusses the Watershed segmentation and clustering results of the two feature schemes in the proposed framework. Details are presented in the following sections.

5.1. Size of the (visual) word dictionary

The number of key points in a dictionary is represented by K (see Eq. (3)), which is also a parameter of the applied k-means clustering. To determine an appropriate value for K, we minimize the ratio between intra-cluster (SSE - sum of squared errors) and inter-cluster (distance to the closet cluster) distances (defined in Eq. (10)) as proposed by Ray and Turi (1999).

$$D^{intra}(=SSE) = \sum_k \sum_i (x_i^{C_k} - c_k)^2$$
$$D^{inter} = \min_{p \neq q}(c_p - c_q)^2$$
$$c_k = \frac{1}{|C_k|} \sum_i x_i^{C_k}$$

(10)

in which, C_k is the k_th cluster and has $|C_k|$ instances, $x_i^{C_k}$ is the i_th instance of C_k, and c_k is the centroid of C_k. In our case, the minimisation of the ratio resulted in $K = 16$.

5.2. Watershed segmentation of congested patterns

As described in the evaluation methodology, we visually analyze the performance of Watershed segmentation (WS) on a variety of congested patterns. Additionally, a qualitative comparison with contour extraction from (Krishnakumari et al., 2017) is provided. Fig. 7a shows examples of congestions from the dataset with different complexity, such as congestion with a single isolated disturbance, multiple sparse disturbances in a congested area and finally, a highly dense congested area. Generally, segmenting these patterns becomes harder as the complexity becomes higher due to the spacing between the disturbances becoming less evident. Corresponding WS results are shown in Fig. 7b. Different regions in a pattern are separated by shed lines, which are indicated in dark blue. The region colours are mapped from the (arithmetic) mean speed of all the interior pixels of a segment. Free flow regions are coloured with the same colour as the shed line since they are referred to as uninterested regions in this study.

We can see that the WS follows and groups pixels in moving jams. Since there is no ground truth for the segmentation, we do not assess the accuracy at the pixel level, but assess the ability of the WS to give distinct disturbances or wide moving jams. In the first pattern in Fig. 7a, WS performs satisfactorily as it can separate the prominent pattern, i.e. the moving disturbance, from the other regions. In the second congestion pattern, several disturbances occur with various sizes and most of them are relatively far apart from each other. The WS result shows that noticeable disturbances are separated clearly. The third pattern is a highly dense congestion with many moving jams being extremely close to each other. Their heads are mixed with the upstream bottleneck which is the potential reason for their emerging. The corresponding WS result presents acceptable segmented regions. Moving jams are tracked and separated correctly eventhough they are sometimes merged together as shown in the third pattern in Fig. 7b. It is clear that tracking and separating these moving jams manually is also difficult. To summarize, segmentation results obtained from WS method show a high ability of separating disturbances in patterns of congestion.

The contour extraction from (Krishnakumari et al., 2017), also termed as naive contour extraction in this paper, is also applied to these patterns to provide a comparison. The main criterion is how well they support the disturbance identification step. The corresponding contour extraction results are shown in Fig. 7d. For the WS, the contours are obtained by extracting the outlines of the corresponding regions as shown in Fig. 7c. It can be observed that, for the simple pattern with one single disturbance, the first one in Fig. 7a, contours obtained from the two methods are able to identify the corresponding disturbance. However, for more complex patterns, naive contour extraction is unable to identify different contours of moving jams. Hence, WS-based contour extraction performs superior to naive contour extraction method and can improve the quality of detecting disturbances.

In summary, some insights can be deduced from the performance of the WS as follows:

- Watershed segmentation is capable of tracking the propagation of a disturbance. It also works well for densely congested areas where multiple moving jams are close to each other.
- Contours extracted based on WS segmentation appear to capture moving disturbances better than that of naive contour extraction.

5.3. Clustering of congested patterns

This section presents the clustering results corresponding to the two feature sets described in the framework - point-based and area-based. They are evaluated using the methodology described in Section 4.3. The hierarchical agglomerative clustering technique results in a hierarchy of data points and is effectively represented by dendrograms, which are binary tree-based representations. The height at which two branches are combined denotes their distance or dissimilarity. Fig. 8 shows two dendrogram plots of the clustering results using the point-based and area-based features respectively.

Upon initial inspection, it can be seen that these two trees exhibit strikingly different properties in at least two perspectives: (i) leaf- combinations, (ii) inter-cluster dissimilarities. Firstly, leaf combinations show how similar any pairs of patterns (i.e. leaves) are. In the point-based tree, most of the patterns connect to fairly long vertical stems; this indicates certain levels of dissimilarities between these patterns. On the other hand, the majority of the (direct) leaf connections in the area-based dendrogram are through extremely short vertical stems, meaning strong similarities between them. (Intuitively, one can easily observe this by looking at the dense parts at the bottom of each of the two dendrograms.) Secondly, regarding (intermediate) cluster differences, the two dendrograms show different properties. One can look at the upper vertical stems, which are not directly connecting leaves, to analyse these inter-cluster dissimilarities. While in the left dendrogram, the lengths of these stems hardly get longer from bottom to top (as clusters are becoming bigger), the right dendrogram shows significant increases of the lengths of these stems. This suggests a higher level of difference between patterns representing by the area-based feature scheme compared to that the point-based scheme. There are two possible hypotheses for these differences in these two dendrograms: the dimensionality of feature vector and the characteristics conveyed by extracted features. In fact, one can expect that the higher the dimension of a feature vector is, the further the distances between patterns is likely to be. In addition, traffic engineers consider phenomena happening in patterns to judge if they are similar. To some extent, this is inlined with the approach of area-based method which looks at patterns at a global, abstract level. The point-based looks into more local features which can possibly find differences between *similar* patterns. Although this does not imply a better clustering of the area-based approach, this explains the correlation of feature characteristics and resulting dendrograms.

Concerning the choice of the number of potential clusters in the dataset, one can cut a dendrogram at a certain horizontal level. From these, separated sub-trees are formed, each with an individual cluster. It is not clear from the point-based dendrogram where to cut the tree since the inter-cluster dissimilarities are mostly much smaller than that of intra-cluster. In contrast, the area-based tree does suggest some potential number of clusters. For instance, we can cut the tree in Fig. 8b by a horizontal line at distance 1.00 to obtain 4 different clusters. The following sections evaluate these clustering results using the proposed qualitative and quantitative methods.

5.3.1. Qualitative

To assess the quality of the clustering results, we consider different (intermediate) clusters, i.e. branches, suggested by the two dendrograms in Fig. 8. The quality of a clustering is measured based on similarities between patterns in the same cluster. Here we visually analyse patterns with respect to their spatio-temporal appearances and traffic knowledge as described in the evaluation methodology.

For the point-based dendrogram, we first look at the two topmost sub-trees (or branches) and see that: While the left sub-tree consists of small, simple patterns of disturbances, the right sub-tree observes a variety of pattens. Therefore, we further look at clusters that (i) are at low distances, meaning high similarities between patterns, and (ii) have a relatively high number of patterns, for instance, clusters with less than ten objects that are not considered. By doing so, we found five types of traffic congestion. Some

(a)   (b)
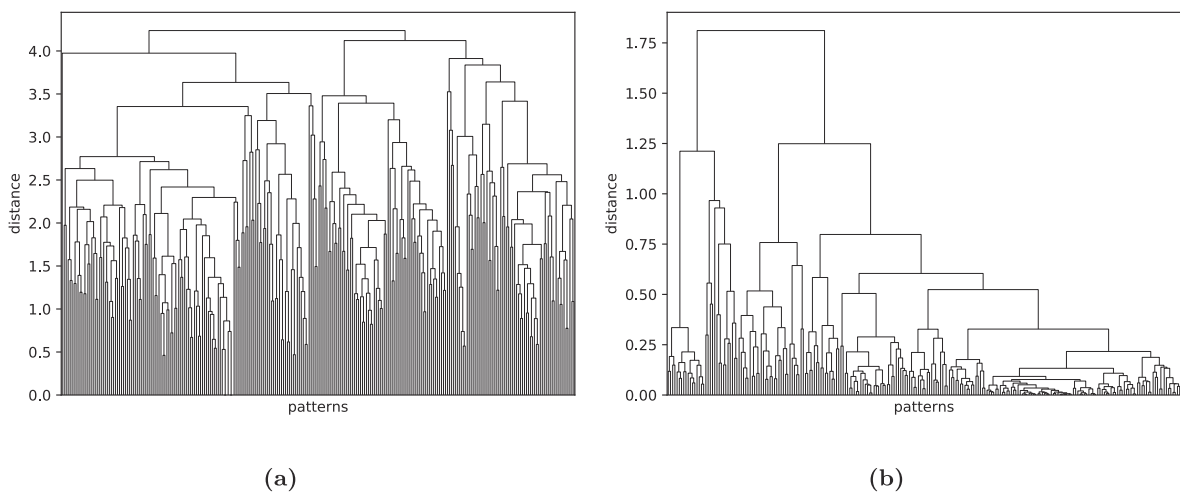
Fig. 8. Dendrogram representations of hierarchical clustering results from different feature schemes: (a) Point-based features and (b) Area-based features. On the horizontal axis lie all the patterns in the dataset. For a clear visualization, the dendrogram organizes all patterns in a way that allows patterns in the same cluster (at any iteration of the hierarchical clustering algorithm) stand next to each other and form a group. (Notice that, there are different orders that satisfy the requirement). The vertical axis shows distances measured between two clusters. One can observe the dendrogram from bottom to top and see when (the order) and which two patterns/clusters are grouped into a bigger cluster.

examples of those are shown in Table 2. Their locations in the dendrogram are highlighted in Fig. 9.
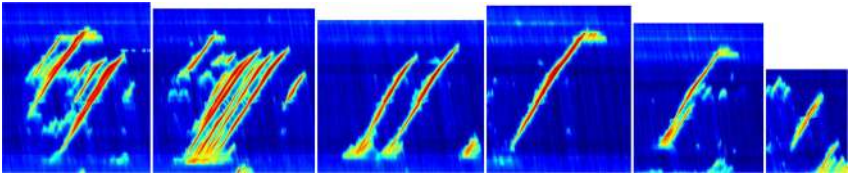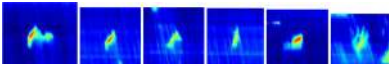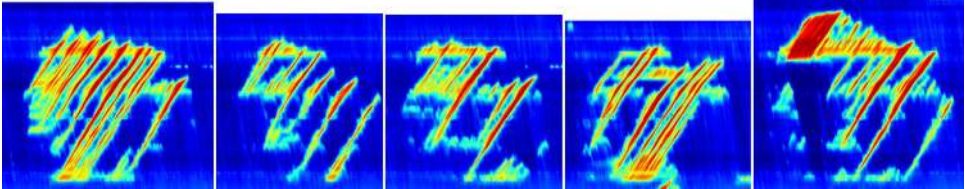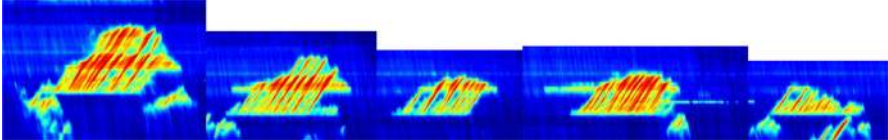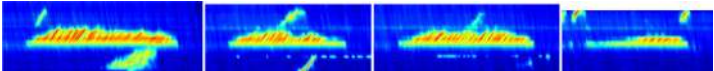
It can be seen from Table 2 that this approach is capable of capturing some typical patterns of congestion. The left (topmost) sub-tree encompasses two clusters of patterns, namely *Moving disturbances (WMJ)* - PC1 and *Small disturbances* - PC2. Most of the patterns in PC1 present single or a very few disturbances spilling back against the driving direction, which are so called wide moving jam (WMJ). The PC2 comprises of very light disturbances that are both spatially and temporally short. In cluster PC3 and PC4, there are many moving disturbances in various combinations. The striking difference between these two is that those in PC4 emerge from fixed bottleneck locations while that is not the case in PC3 where origins of disturbances are at different locations. Therefore, to discern between these patterns, we call them *High frequency of WMJs* and *High frequency of WMJs emerging from bottlenecks*, respectively. The last cluster (PC5) shows traffic congestion that is stationary at fix bottlenecks for long periods; hence, we name it *Bottleneck*.

Despite the four typical patterns that have just been described, there are some limitations of this approach as follows. Firstly, the clusters listed in Table 2 are sub-trees of the corresponding dendrogram (as highlighted in Fig. 9), which in aggregation does not cover the whole dataset. It takes much effort, e.g. trial-and-error, to explore and locate them in the dendrogram. Secondly, from a traffic knowledge point of view, similar patterns are expected to be grouped at as much lower level (of distances). This requirement is not met with the point-based results. For instance, the PC2 consists of two (yellow) sub-trees which are not directly connected with each other even though they describe the same pattern. Moreover, there are many patterns such as WMJs and small disturbances in black colour that are not partitioned to their corresponding PC1 and PC2. Their unexpected positions are rather hard to explain; one might need to investigate their extracted key points to do so. As shown previously, key points essentially captured characteristics of neighbourhoods surrounding particular locations in images. Additionally, the construction of a dictionary of visual words disregards potential spatial or temporal correlations of these points and only considers their occurrence frequencies. That means, to some extent, losing potentially valuable information is inevitable.

For the area-based approach, we also examine patterns from various (intermediate) clusters. The obtained result include five clusters with strong intra-cluster similarities. Table 3 presents some examples from these clusters together with their sizes. Additionly, their corresponding locations are highlighted in the dendrogram in Fig. 10. It can be seen from the dendrogram that these clusters encompass the whole dataset. Statistically, the smallest cluster - AC5, has 15 patterns while most of the patterns, particularly 154, are categorized into the first cluster (AC1). This means AC1 is the most recurrent congestion found in the A12 road segment for the four studied months.

Table 2
Description of point-based clustering result. These clusters are corresponding with clusters shown in the dendrogram in Fig. 9. It should be noticed that they are represented in a deliberate order to facilitate the corresponding discussion.

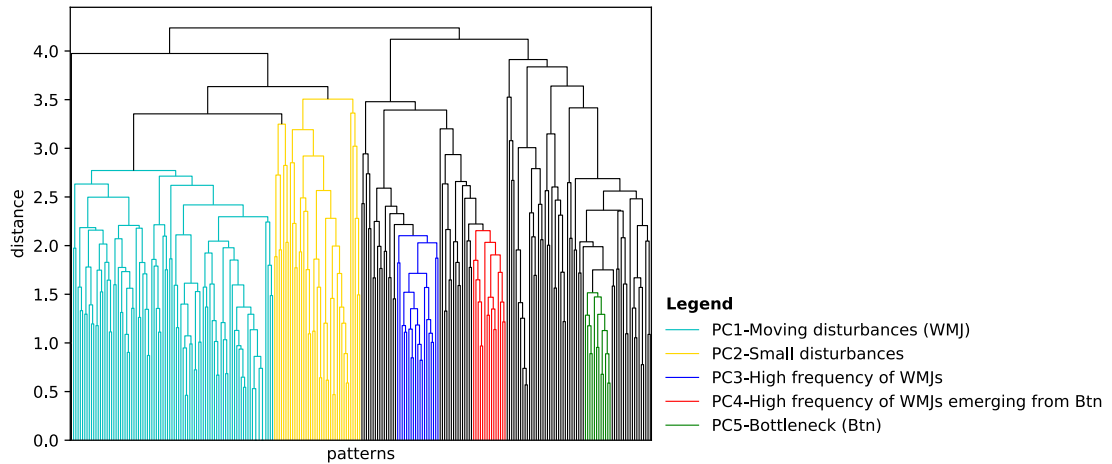| Cluster	Number of patterns	Examples
PC1	90	
PC2	39	
PC3	19	
PC4	15	
PC5	13	

Fig. 9. Visualization of four typical congestion patterns found in the dataset by applying hierarchical agglomerative clustering on the point-based feature scheme.

Table 3
Description of area-based clustering result. These clusters are corresponding with clusters shown in the dendrogram in Fig. 10. They are ordered deliberately to facilitate the corresponding discussion.

Cluster	Number of patterns	Examples
AC1	154	
AC2	30	
AC3	16	
AC4	17	
AC5	15	

The first cluster (AC1) shows congestion with one or a very few disturbances. Most of them describe so-called wide moving jam with the minor are insignificant disturbances, see examples in Table 3. We therefore name this cluster *disturbance*. The second cluster (AC2) shows patterns with a number of moving disturbances. Consequently, a reasonable label for this cluster is *high frequency of*
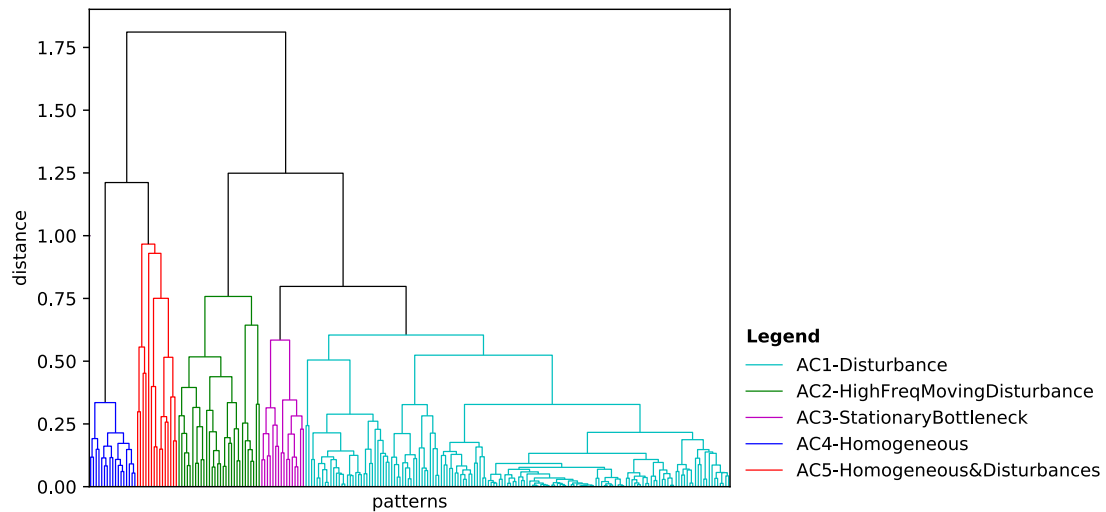
Fig. 10. Visualization of five typical congestion patterns found in the A12 dataset by applying hierarchical agglomerative clustering on the area-based feature scheme.

disturbances. It is noticeable that most of the patterns in the first two clusters constitute of moving disturbances. The third cluster (AC3) illustrates *bottleneck* congestion as described in the point-based approach. Patterns in cluster AC4 are small scaled both spatially and temporally compared to those in other clusters. Furthermore, they mainly show the demand-supply patterns, which cause slow traffic. Since traffic presented inside these patterns is homogeneous, a suitable label for this cluster is *small scaled homogeneous congested traffic*. Lastly, the fifth cluster (AC5) shows complex patterns with compounds of homogeneous congestions and moving disturbances. Additionally, their spatio-temporal scale is large. Hence, these are defined as *Homogeneous* & *Disturbances* clusters.

The dendrogram in Fig. 10 can lead to some insights into similarity scores within each of the five clusters. Cluster AC4, in blue, has the lowest distance, meaning lowest variance. The construction of the feature vector is likely to account for this minimal intra-cluster dissimilarity. Firstly, the demand-supply elements are logically indicated as existing in these patterns, regardless of its properties (such as shape or scale). Secondly, there are very few, if not no, disturbances in these patterns. Thirdly, the congestion is in small scale. These three characteristics lead to a minimal distance between patterns in cluster AC4. Intra-cluster distances of AC3 and AC4 are more or less at the same level. AC2 is a bit higher in comparison and AC5 is the most diverse one. The complexity of *Homogeneous* & *Disturbance* patterns - large differences in scales and various number of disturbances (see examples in Table 3) - appears to account for this highest distance in AC5.

It is noteworthy that obtaining these five clusters by cutting the dendrogram with one horizontal line is not possible. This situation is caused by high levels of dissimilarity of patterns in the compound pattern cluster. To automate the process of getting these five clusters, one can define a hierarchical approach as follows. (1) The first step is to cut the dendrogram into two clusters, which are the two branches with and without demand-supply elements in congestion patterns. Let us call them CisDS and CnoDS respectively. (2) The second step deals with these clusters separately. The CisDS is cut into two branches, which are AC4 and AC5. The CnoDS is cut into three branches which are AC1, AC2, AC3. An alternative approach can be a manipulation of the distance function or feature weighting to control the distance between AC5 patterns. However, we do not explore this further in this study.

A comparison between these two clustering results can be made from two perspectives. First, regarding classification application, the area-based approach successfully groups patterns into corresponding categories as its five clusters cover the whole dataset. In contrast, the four clusters from the point-based approach only capture a part of the dataset. Second, from a viewpoint of typical pattern recognition, both approaches show clusters of *bottlenecks* and *WMJs*. We ignore the fact that the point-based approach can miss some of these patterns and rather focus on the typical emerging patterns. While the area-based approach group all the High frequency of WMJs together, the point-based approach can separate them into those that originate from bottlenecks and those that do not. The area-based approach can identify and separate demand-supply patterns into simple ones and compound ones that are most likely to be omitted by the point-based approach. This comparison suggests that the area-based approach is applicable for the classification of congestion patterns at a high abstraction level, and the point-based approach tends to work well with a lower detail level. This is well aligned with the methodology adopted in these two approaches. Hence, one can hierarchically start with the area-based approach to obtain the five clusters and use the point-based approach to partition the High frequency of WMJs further, for instance.

Interestingly, the five clusters resulting from the area-based method coincide largely with the five manually selected labels in previous work (Krishnakumari et al., 2017; Nguyen et al., 2016) (as mentioned in Section 2). It can be seen that these classes are well captured with the clusters in Table 3. The first two classes (WMJ + low frequency of WMJs) are generally grouped in cluster one, high frequency of WMJs in cluster two, homogeneous congestion in cluster four and mixed class cluster in cluster five. It should be noted that dividing traffic oscillation into low and high frequency is somewhat subjective due to the decision of threshold number of moving jams. There is a difference to the approach described here; stationary bottleneck patterns are clustered by the area-based method which is not considered in Krishnakumari et al. (2017) and Nguyen et al. (2016).
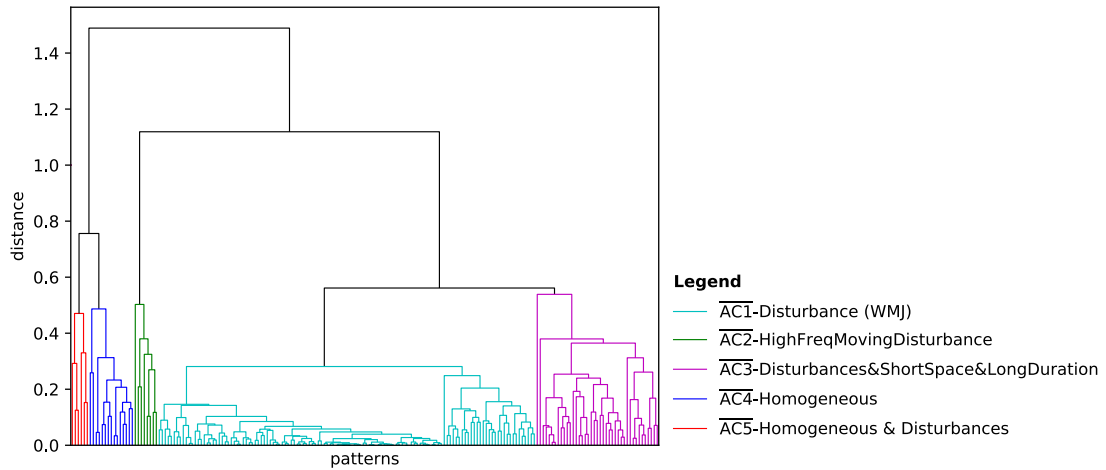
Fig. 11. Visualization of five typical congestion patterns found in the A13 dataset by applying hierarchical agglomerative clustering on the area-based feature scheme.

The area-based clustering results are also in line with the classification in Helbing et al. (2009) that we discussed in Section 2. Our C1 cluster matches with the moving cluster C1. Both SGW and OCT comprise a frequent occurrence of moving jams, which can also be represented by cluster AC2. As noticed by the authors in Helbing et al. (2009), differences between these two types are not straightforward. However, one might suggest to depend on the (temporal) distances between moving jams to discern them. Subsequently, introducing further variables to the feature vector such as density of moving jams over time, might separate AC2 into SGW and OCT. PLC is essentially recognized by cluster AC3 and HCT by AC4. The WSP is not recognized in this result. This is likely due to the choice of the low-speed threshold, which might discard these patterns.

In order to test the transferability of the area-based method, we apply it to clustering of congestion patterns in the A13 dataset. The resulting dendrogram has a similar structure to that obtained from A12 dataset, from which five clusters are identified and highlighted, as shown in Fig. 11. Four of them are comparable with those found in A12 dataset: disturbances, high frequencies of disturbances, homogeneous and mix of homogeneous and disturbances. The cluster $\overline{AC3}$ consists of bottleneck related patterns like those in cluster AC3 (A12 dataset); however, it also includes patterns that have many disturbances but with spatially short extents. Hence, it appears that short space, long duration and having disturbances are the common features of patterns in $\overline{AC3}$. Some of these patterns can belong to $\overline{AC2}$. This is possibly because that although the area-based feature vector includes disturbances and scales of patterns, it ignores the spatial length of such disturbances. One might include this information (disturbance lengths), in an appropriate approach, to distinguish spatially short disturbances in bottlenecks with long moving disturbances in $\overline{AC2}$. Nonetheless, the result from A13 dataset indicates that the area-based approach is transferable and able to cluster a dataset into meaningful groups.
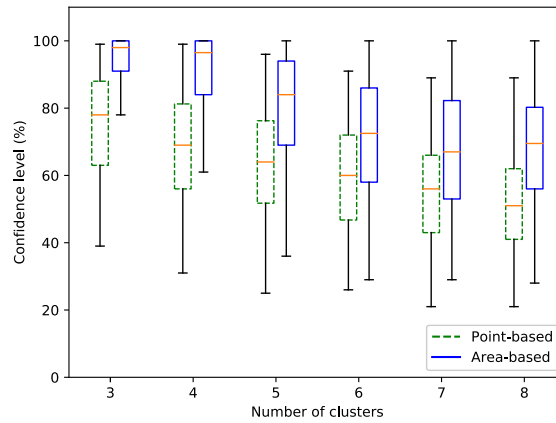
5.3.2. Quantitative

We apply the evaluation methodology described in Section 4.3.2 with the following settings: 100 folds ($N = 100$) of the dataset and number of clusters varying from three to eight. We collect every confidence level of the classifier when assigning class labels to patterns; those levels, calculated by the classifier, are probabilities associating with corresponding labels. Box plots are used to present the distribution of these levels of confidence in each option of the number of clusters. Fig. 12 shows the final result.

Overall, a general declining trend is observed for both methods. This tendency is generally expected since the problem becomes more challenging when the dataset is partitioned into a higher number of clusters. In fact, the corresponding classifier becomes less confident in its decisions when more classes are involved. Additionally, given the same features, the separability of clustering decreases when the number of clusters increase. This leads to the decrease in the confidence level of the classifier. Fig. 12 also shows that the result of the area-based approach is superior to that of the point-based approach because of the higher locations of the rectangle boxes, which encapsulate the central 50% population of confidence values. Additionally, the median confidence levels of the area-based scheme are considerably higher than that of the point-based scheme over all the choices of the number of clusters. The highest median confidence level observed with the point-based approach is approximately 80 with 3 clusters. Additionally, the confidence level drops to around 50% for larger number of clusters, which indicates a high confusion level of the classifier. On the other hand, all the median confidence levels in the area-based approach are higher than 65%. Three of them are higher than 80%, hence, higher than the highest value in the other approach. This indicates that the confidence of classifiers learned from area-based clustering is strong compared to the point-based. Hence, it can be concluded that the area-based approach results in a higher level of separability of the data set than the point-based approach.

6. Conclusions

This paper explores two methods to automatically label and separate traffic patterns that come in the form of images (heatmaps of speeds over space and time along a road corridor). Both methods have their origins in image processing and conceive spatio-temporal

Fig. 12. Quantitative evaluation results of the area-based and point-based approaches. Each box plot includes following elements: short horizontal bars, so-called whisker, at top and bottom extremes indicate maximum and minimum values respectively; the lower bound of each rectangular is at 25 percentile and the upper bound at 75 percentile; the horizontal red line inside the rectangle of each box plot indicates the median value.

traffic patterns as images where each pixel represents speed in a particular space-time area. The point-based (bag-of-feature) approach explores key points - locations - in traffic (image) patterns. These points do not necessarily have a direct meaning related to traffic. The area-based method uses the Watershed operator to segment a pattern into different areas from which three main traffic related elements are extracted—spatial and temporal scales, disturbances and (heavily congested) demand-supply areas. In this respect, this second *new* approach is a hybrid method that combines image processing and (traffic) domain knowledge. Hierarchical agglomerative clustering is performed on the feature sets derived from both methods which results in different hierarchies of the data set.

Since labelling traffic (congestion) patterns is highly subjective, we test the methods both *qualitatively*, (does the method result in meaningful clusters/ labels?) and *quantitatively* (does the method separate the resulting feature space such that we can have high confidence in the results when applying a classifier using the labels on (large amounts of) unseen (data)?) From the findings, we conclude the following:

- Both qualitatively and quantitatively, the hybrid area-based method is superior to the point-based method. The resulting labels of the first method are more intuitive and explicable (e.g. we see clusters with homogeneous congestion; isolated wide moving jams and high complex mixed patterns); and the separability of the resulting feature space is excellent over the entire range of number of clusters tested.
- More specifically:
  - The area-based method results in (by design) directly interpretatable features (e.g. disturbances, heavily congested regions, total spatio-temporal extent of congestion)
  - The obtained segmentation provides a way to process different elements in the corresponding pattern separately and allows various features to be derived. This may be beneficial in wider applications, e.g., in the identification of wave speeds or shock fronts from the segmented regions.
  - The area-based method yields a much smaller feature space and is thus (much) more parsimonious than the point based method.

However, our results do not suggest one should dismiss the point-based approach altogether. In contrast, our result highlight its capability for recognizing more detailed and subtle differences between clusters (labels). This suggests a hierarchical approach to clustering data sets may be possible:

- The area-based approach may be used as a top level approach to partition the data set based on domain knowledge.
- Subsequently, the point-based approach could be employed to further discriminate between patterns in more detail, without a priori conceptions about what these differences (physically) entail. By doing this, traffic patterns can be annotated with meaningful labels at both abstract and detail levels which potentially describe them better.

These latter bullets indicate one important line of further research (hierarchical and mixed clustering, labelling and classification). A second line of further research lies in the embedding of these methods in an overall (database) search engine that uses the reduced feature space to query the (fast growing) 200 TB traffic database of the Dutch National Datawarehouse and for fast retrieval, visualisation and analysis of traffic patterns. We finally highlight some important limitations in this work. In this study, only a limited number of features are used in area-based approach; there are still opportunities for further improvement in this direction. Adding (extracting) more traffic-related features can be used for exploring typical patterns from different perspectives. Furthermore, the ability to recognize new traffic patterns is also an important topic for future research. In this respect, we seek a classifier that learns new patterns, without "forgetting" existing patterns.

Acknowledgements

References

A12 highway. <https://en.wikipedia.org/wiki/A12_motorway_(Netherlands)>.

A13 highway. <https://en.wikipedia.org/wiki/A13_motorway_(Netherlands)>.

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-up robust features (surf). Comput. Vision Image Understand. 110 (3), 346–359.

Beucher, S., Meyer, F., 1992. The morphological approach to segmentation: the watershed transformation. Opt. Eng.-New York-Marcel Dekker Incorporated- 34, 433.

Böhning, D., 1992. Multinomial logistic regression algorithm. Ann. Inst. Stat. Math. 44 (1), 197–200.

Canny, J., 1986. A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. (6), 679–698.

Celikoglu, H.B., 2013. An approach to dynamic classification of traffic flow patterns. Comput.-Aided Civil Infrastruct. Eng. 28 (4), 273–288.

Celikoglu, H.B., Silgu, M.A., 2016. Extension of traffic flow pattern dynamic classification by a macroscopic model using multivariate clustering. Transp. Sci. 50 (3), 966–981.

Chui, C.K., 2016. An Introduction to Wavelets. Elsevier.

Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J., 1995. Active shape models-their training and application. Comput. Vision Image Understand. 61 (1), 38–59.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297.

Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. IEEE Trans. Inf. Theory 13 (1), 21–27.

Depaire, B., Wets, G., Vanhoof, K., 2008. Traffic accident segmentation by means of latent class clustering. Acc. Anal. Prevent. 40 (4), 1257–1266.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S., Bouras, A., 2014. A survey of clustering algorithms for big data: taxonomy and empirical analysis. IEEE Trans. Emerg. Top. Comput. 2 (3), 267–279.

Gao, J., Andrew, G., Johnson, M., Toutanova, K., 2007. A comparative study of parameter estimation methods for statistical natural language processing.

Haralick, R.M., Sternberg, S.R., Zhuang, X., 1987. Image analysis using mathematical morphology. IEEE Trans. Pattern Anal. Mach. Intell. (4), 532–550.

Harris, C., Stephens, M., 1988. A combined corner and edge detector. In: Alvey Vision Conference, vol. 15, Citeseer, pp. 10–5244.

Helbing, D., Hennecke, A., Treiber, M., 1999. Phase diagram of traffic states in the presence of inhomogeneities. Phys. Rev. Lett. 82 (21), 4360.

Helbing, D., Treiber, M., Kesting, A., Schönhof, M., 2009. Theoretical vs. empirical classification and prediction of congested traffic states. The Eur. Phys. J. B-Conden. Matter Complex Syst. 69 (4), 583–598.

Huang, F.-L., Hsieh, C.-J., Chang, K.-W., Lin, C.-J., 2010. Iterative scaling and coordinate descent methods for maximum entropy models. J. Mach. Learn. Res. 11 (Feb), 815–848.

Jain, A.K., 2010. Data clustering: 50 years beyond k-means. Pattern Recogn. Lett. 31 (8), 651–666.

Kadir, T., Brady, M., 2001. Saliency, scale and image description. Int. J. Comput. Vision 45 (2), 83–105.

Kerner, B.S., 2002. Empirical macroscopic features of spatial-temporal traffic patterns at highway bottlenecks. Phys. Rev. E 65 (4), 046138.

Kerner, B.S., Rehborn, H., Aleksic, M., Haug, A., 2004. Recognition and tracking of spatial–temporal congested traffic patterns on freeways. Transp. Res. Part C: Emerg. Technol. 12 (5), 369–400.

Kim, J., Mahmassani, H.S., 2015a. Spatial and temporal characterization of travel patterns in a traffic network using vehicle trajectories. Transp. Res. Part C (59), 375–390.

Kim, J., Mahmassani, H.S., 2015. Trajectory clustering for discovering spatial traffic flow patterns in road networks, Tech. rep.

Krishnakumari, P., Nguyen, T., Heydenrijk-Ottens, L., Vu, H.L., van Lint, H., 2017. Traffic congestion pattern classification using multiclass active shape models. Transp. Res. Rec.: J. Transp. Res. Board (2645), 94–103.

Lee, H., Lee, H.-W., Kim, D., 2000. Phase diagram of congested traffic flow: an empirical study. Phys. Rev. E 62 (4), 4737.

Lowe, D.G., 1999. Object recognition from local scale-invariant features. In: The proceedings of the seventh IEEE international conference on omputer vision, 1999, vol. 2. IEEE, pp. 1150–1157.

National datawarehouse of traffic information. <http://www.ndw.nu/en/>.

Nguyen, H.N., Krishnakumari, P., Vu, H.L., van Lint, H., 2016. Traffic congestion pattern classification using multi-class svm. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 1059–1064.

Rambabu, C., Chakrabarti, I., 2008. An efficient hillclimbing-based watershed algorithm and its prototype hardware architecture. J. Signal Process. Syst. 52 (3), 281–295.

Ray, S., Turi, R.H., 1999. Determination of number of clusters in k-means clustering and application in colour image segmentation. In: Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques, Calcutta, India, pp. 137–143.

Rendón, E., Abundez, I., Arizmendi, A., Quiroz, E.M., 2011. Internal versus external cluster validation indexes. Int. J. Comput. Commun. 5 (1), 27–34.

Schönhof, M., Helbing, D., 2007. Empirical features of congested traffic states and their implications for traffic modeling. Transp. Sci. 41 (2), 135–166.

Schreiter, T., van Lint, H., Treiber, M., Hoogendoorn, S., 2010. Two fast implementations of the adaptive smoothing method used in highway traffic state estimation. In: 2010 13th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 1202–1208.

Sobel, I., 1990. An isotropic 3× 3 image gradient operator. Mach. Vision Three-Dimensional Scenes 376–379.

Soriguera, F., Robusté, F., 2011. Estimation of traffic stream space mean speed from time aggregations of double loop detector data. Transp. Res. Part C: Emerg. Technol. 19 (1), 115–129.

Spiliopoulou, A., Kontorinaki, M., Papageorgiou, M., Kopelias, P., 2014. Macroscopic traffic flow model validation at congested freeway off-ramp areas. Transp. Res. Part C: Emerg. Technol. 41, 18–29.

Treiber, M., Helbing, D., 2002. Reconstructing the spatio-temporal traffic dynamics from stationary detector data. Cooper. Transp. Dyn. 1 (3), 1–3.

Van Lint, J., Hoogendoorn, S.P., 2010. A robust and efficient method for fusing heterogeneous data from traffic sensors on freeways. Comput.-Aided Civil Infrastruct. Eng. 25 (8), 596–612.

Van Lint, J., Van Zuylen, H.J., Tu, H., 2008. Travel time unreliability on freeways: why measures based on variance tell only half the story. Transp. Res. Part A: Policy Pract. 42 (1), 258–277.

Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2005. Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. Transp. Res. Part C: Emerg. Technol. 13 (3), 211–234.

Wang, Y., Papageorgiou, M., Messmer, A., 2006. Renaissance–a unified macroscopic model-based approach to real-time freeway network traffic surveillance. Transp. Res. Part C: Emerg. Technol. 14 (3), 190–212.

Witten, I.H., Frank, E., Hall, M.A., Pal, C.J., 2016. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann.

Xu, D., Tian, Y., 2015. A comprehensive survey of clustering algorithms. Ann. Data Sci. 2 (2), 165–193.

Xu, R., Wunsch, D., 2005. Survey of clustering algorithms. IEEE Trans. Neural Networks 16 (3), 645–678.

Yu, H.-F., Huang, F.-L., Lin, C.-J., 2011. Dual coordinate descent methods for logistic regression and maximum entropy models. Mach. Learn. 85 (1–2), 41–75.

Zheng, N., Waraich, R.A., Axhausen, K.W., Geroliminis, N., 2012. A dynamic cordon pricing scheme combining the macroscopic fundamental diagram and an agent-based traffic model. Transp. Res. Part A: Policy Pract. 46 (8), 1291–1303.