Neural Computing and Applications (2022) 34:18545–18560
https://doi.org/10.1007/s00521-022-07433-9

ORIGINAL ARTICLE

Analysis of moving cluster with scene constraints for group behavior pattern mining

Yuanfeng Yang[1,2] · Lin Li[3] · Liang Zhang[2] · Gang Liu[2] · Zhaobin Liu[2]

Received: 17 December 2021 / Accepted: 11 May 2022 / Published online: 19 June 2022

Abstract
Group behavior pattern mining in traffic scenarios is a challenging problem due to group variability and behavioral regionality. Most methods are either based on trajectory data stored in static databases regardless of the variability of group members or do not consider the influence of scene structures on behaviors. However, in traffic scenarios, information about group members may change over time, and objects' motions show regional characteristics owing to scene structures. To address these issues, we present a general framework of a moving cluster with scene constraints (MCSC) discovery consisting of semantic region segmentation, mapping, and an MCSC decision. In the first phase, a hidden Markov chain is adopted to model the evolution of behaviors along a video clip sequence, and a Markov topic model is proposed for semantic region analysis. During the mapping procedure, to generate snapshot clusters, moving objects are mapped into the corresponding sets of moving objects according to the semantic regions where they are located at each timestamp. In the MCSC decision phase, a candidate MCSC recognition algorithm and screening algorithm are designed to incrementally identify and output MCSCs. The effectiveness of the proposed approach is verified by experiments carried out using public road traffic data.

Keywords Moving cluster with scene constraints · Group pattern · Behavior understanding · Scene modeling · Semantic region

1 Introduction

With the development of computer vision and network technology, the intelligent video surveillance system has become an important part of an intelligent transportation system (ITS). Video sensor nodes enable the timely and accurate capture of the visual appearance of moving objects, and the extraction of information about them, such as through object detection, tracking, and behavior analysis. The research focus of intelligent video surveillance is shifting to the understanding of behavior patterns extracted from traffic video data using machine learning methods, which has attracted attention in the field of computer vision [1].

Current research on behavior understanding includes the steps of establishing behavior models from training samples and matching test samples with established models [2, 3], where typical behavior models are obtained by clustering trajectory data. In traffic monitoring scenarios, combined with scene knowledge, semantic interpretation of behaviors can be given, such as heading straight, turning left, turning right, making a U-turn, reverse driving, and illegal lane changing [4–7]. These are classified as individual patterns (regular patterns of an individual) according to the granularity of trajectory patterns [8]. Individual behaviors are identified by matching with established behavior models, whose training samples generally come from static datasets, which cannot reflect real-time patterns of behavior and incrementally output results.

✉ Yuanfeng Yang
yangyuanfeng@jssvc.edu.cn

[1] Jiangsu Province Support Software Engineering R&D Center for Modern Information Technology Application in Enterprise, Suzhou 215104, China

[2] School of Computer Engineering, Suzhou Vocational University, Suzhou 215104, China

[3] School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361005, China

The discovery of group behavior patterns differs fundamentally in that the patterns of discovery follow a given notion of common patterns of different objects. These can be exploited to describe traffic scenarios in real time, answer group behavior questions such as “when and where do the aggregation and segregation of objects’ movements?” and “what are the traffic behavior rules in scenarios at the current time?” Through the identification of these group behavior patterns, the rules and trends of group movements and the occurrence of various mass incidents can be more effectively reflected, so as to better manage traffic infrastructure. However, these problems cannot be resolved by extracting individual patterns.

Studies have been proposed to discover group patterns [9–18], whose common point, such as a group [9], flock [10], convoy [11], swarm [12], traveling companion [13], or loose traveling companion [14], is that the objects’ groups are required to contain the same collection of moving objects. However, this does not always conform to the actual situation. In a monitoring scene, within a certain time interval, the set of objects continually changes as they enter or exit. In addition, due to regional characteristics shown by group behavior patterns (such as convergence and divergence), moving objects may pass through different combinations of semantic regions, which correspond to different behavior categories. The sets of moving objects within these clusters change over time, while a trajectory cluster has a constant set of objects regardless of their temporal information. It is critical that all the objects’ motions are constrained by scene structures. Based on this, we define moving clusters with scene constraints (MCSCs), where two consecutive snapshot clusters share many common objects. Hence, the above group patterns [9–14] are unsuitable to model an MCSC.

Other kinds of group patterns also exist, such as the moving cluster [15], gathering [16, 17], or evolving convoy [18], which have no strict requirements that objects’ groups contain the same moving objects, allowing members to leave a group during its life cycle, while new members can join it. To retrieve these group patterns, density-based clustering algorithms such as DBSCAN [19] and OPTICS [20] are adopted to cluster moving objects at each timestamp. However, in traffic monitoring scenarios, moving behaviors of vehicles are related to semantic regions passing by, but not to vehicle density. Density-based clustering algorithms have difficulty identifying group movements of vehicles.

Existing methods cannot effectively support the discovery of MCSCs, because of challenges including the following:

(1) Group variability: The membership information of MCSCs may change over time. However, many state-of-the-art trajectory-based approaches and group discovery methods require the group to contain a constant set of objects, ignoring changes in group membership;
(2) Behavioral regionality: Due to scene structures, objects’ motions in the same semantic region have similar semantic interpretations. However, clusters identified by density-based clustering algorithms, which are adopted by most group discovery methods at each timestamp, do not always consist of objects in the same semantic region. Therefore, they cannot be directly used for MCSC discovery;
(3) Incremental discovery: To describe traffic scenarios and answer group behavior problems in real time, MCSCs should be reported incrementally, i.e., the discovery algorithm should output the results while processing the trajectory data flow.

We propose a framework to extract meaningful group behavior patterns from traffic video data. Using a hidden Markov chain to model the evolution of behaviors along a video clip sequence, the Markov topic model (MTM) learns the distribution of topics and maps the learned topics to semantic regions in a scene. Under our framework, video clips are treated as documents, and trajectory points (positions and moving directions) of moving objects as motion words. Topics model semantic regions, which correspond to common paths, are taken by objects in a scene. At each timestamp, instead of clustering moving objects using density-based clustering algorithms, moving objects passing through the same semantic region are mapped to the corresponding moving object sets to generate snapshot clusters, which is more consistent with the actual movement situation of vehicles in traffic scenarios. In the MCSC decision phase, at each timestamp, lists of candidate MCSCs are constructed by a candidate MCSC recognition algorithm, and MCSCs in the candidate lists are identified and output incrementally by a screening algorithm, which effectively reduces the number of redundant MCSCs.

The remainder of this paper is organized as follows. Section 2 discusses related work, and Sect. 3 states the problem of MCSCs in traffic scenarios. A Markov topic model is introduced for semantic region analysis in Sect. 4. Section 5 describes discovery algorithms of MCSC patterns. Section 6 presents our experimental results, and we discuss our conclusions and suggest future work in Sect. 7.

2 Related work

2.1 Behavior understanding based on trajectory analysis

Trajectory analysis is the most popular method for vehicle behavior understanding in traffic video surveillance. After constructing a trajectory similarity/distance matrix based on the computed similarity/distance between trajectories, clustering algorithms such as spectral clustering [21, 22], fuzzy C-means [23], and agglomerative hierarchical clustering [24] can be used to group the objects' motion trajectories into different behavior categories. Such unsupervised trajectory pattern classification methods have been extensively used. Other learning methods of trajectory patterns do not need to calculate pairwise similarities between trajectories. GMMs [25], HMMs [26], and self-organizing neural networks [27, 28] can also be used to model the trajectory distribution patterns of moving objects for atypical event recognition and behavior prediction.

The above methods model behaviors with the whole trajectory. However, they may ignore common sub-trajectories that are useful in many applications. Lee et al. [29] proposed a partition-and-group framework that partitions a trajectory into a set of line segments according to the minimum description length (MDL) principle and clusters them using a density-based line-segment clustering algorithm to discover the common sub-trajectories. However, the presented trajectory clustering algorithm TRACLUS is sensitive to the input parameters ε and MinLns, and the trajectory data model ignores the time dimension. Piciarelli et al. [30] proposed a tree-like structure to represent trajectories as a tree of clusters, during whose creation the trajectories are divided into sub-trajectories due to splits. Sub-trajectories are clustered to generate path regions with semantically similar descriptions. A path, represented by all pixels covered by objects with the same behavior cluster along their course, is conceptually related to the proposed semantic region, which is defined as a subset of a path [31]. However, the tree-like structure does not inherently provide a probabilistic interpretation for behavior analysis, and it cannot handle group moving behaviors.

2.2 Learning behavior with topic models

Behaviors are closely related to scene structures, since objects' motions are constrained by them. Wang et al. [31] employed dual-HDP to segment semantic regions, which correspond to the common paths taken by objects in a scene, for activity analysis from trajectories. Li et al. [32–34] segmented monitoring scenarios into multiple semantic regions according to the distribution of low-level atomic video events, where scene segmentation was regarded as an image segmentation problem. The object was for behaviors in each segmented semantic region to have similar characteristics and to be represented by some kind of atomic video events. Zhou et al. [35] proposed a random field topic (RFT) model including sources and sinks as high-level semantic priors to learn semantic regions from tracklets (fragments of trajectories). Similar to Wang et al. [31], these semantic regions corresponded to common paths taken by objects, whose motions in the same semantic region have similar semantic interpretations.

Traffic scenarios include many kinds of behaviors that evolve according to a certain temporal order of traffic flow. The topic model itself ignores the time factor and cannot model the temporal relationships between behaviors. We use a hidden Markov chain to model the temporal correlation between behaviors and propose an MTM to model semantic regions to generate snapshot clusters.

3 Problem statement

Let $T = \{t_1, t_2, \ldots, t_n\}$ be the set of all timestamps, and $O = \{o_1, o_2, \ldots, o_m\}$ the set of all moving objects that have moved during T in the database. Moving object o_j has its own lifetime T_{o_j} from entering to leaving the scene. A snapshot S_i of T is a subset of O, which is located at t_i. The concept of a snapshot is employed as the projection of moving objects' spatial information at one timestamp. The regularity of objects' motions can be reflected by the concurrence of trajectories of the same category (similar moving directions and adjacent spatial positions) in the spatial regions. Such regions are defined as semantic regions.

Let $C = \left\{C^1, C^2, \ldots, C^K\right\}$ be the set of the learned semantic regions in the monitoring scene, which is viewed as the set of general classes. K represents the number of semantic regions. Then, at each timestamp, moving objects are mapped to a certain general class according to the semantic regions in which they are located. Ultimately, the set of snapshot clusters $C_{t_i} = \left\{C_{t_i}^1, C_{t_i}^2, \ldots, C_{t_i}^K\right\}$ at timestamp t_i is generated, where $C_{t_i}^j = \{o_{j1}, o_{j2}, \ldots, o_{jv}\}$ $(1 \le j \le K, 1 \le v \le m)$ is a snapshot cluster at timestamp t_i and m is the total number of moving objects.

An MCSC is a sequence of snapshot clusters at consecutive timestamps, whose definition can be generalized as follows.

Definition 1 Let $g = C_{t_\chi}^{j_\chi}, C_{t_{\chi+1}}^{j_{\chi+1}}, \ldots, C_{t_\psi}^{j_\psi} \left(\forall C_{t_i}^{j_i} \in C\right)$ be a sequence of snapshot clusters, $C_{t_i}^q$ the snapshot cluster at the next timestamp of $C_{t_{i-1}}^p$, and θ the coincidence rate

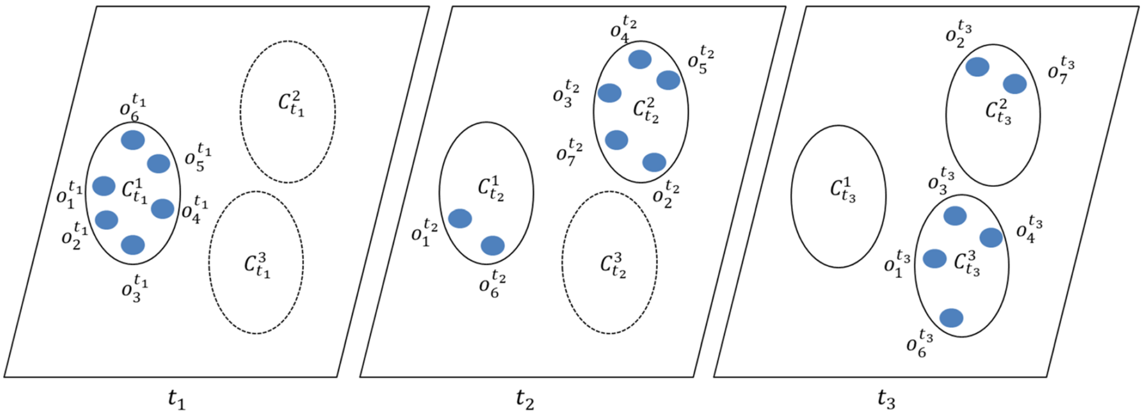


Fig. 1 Example of MCSC

threshold. If $\frac{\left|C^p_{t_{i-1}} \cap C^q_{t_i}\right|}{\left|C^p_{t_{i-1}} \cup C^q_{t_i}\right|} \geq \theta (0 < \theta \leq 1, \chi < i \leq \psi, 1 \leq j_\chi \leq p,$ $q \leq j_\psi \leq K)$, the sequence of snapshot clusters g is defined as an MCSC.

Figure 1 shows an example of an MCSC. There are three semantic regions in a monitoring scene, and their spatial locations and shapes remain constant at each timestamp. Let t_1, t_2, and t_3 be three consecutive timestamps. At each timestamp is a set of snapshot clusters $\left\{C^1_{t_i}, C^2_{t_i}, C^3_{t_i}\right\}$, corresponding to the set of general classes $\left\{C^1, C^2, C^3\right\}$ mapped to the semantic regions in the scene, and $o_j^{t_1}$, $o_j^{t_2}$, and $o_j^{t_3}$ are the representations of moving object o_j at these timestamps. Note that object o_7 enters the scene at timestamp t_2 and object o_5 leaves the scene at t_3. In other words, the membership information of MCSCs may change over time.

Let $\theta = \frac{1}{4}$. $C^1_{t_1} C^2_{t_2} C^3_{t_3}$ is an MCSC, since $\frac{\left|C^1_{t_1} \cap C^2_{t_2}\right|}{\left|C^1_{t_1} \cup C^2_{t_2}\right|} = \frac{4}{7}$ and $\frac{\left|C^2_{t_2} \cap C^3_{t_3}\right|}{\left|C^2_{t_2} \cup C^3_{t_3}\right|} = \frac{2}{7}$ are both at least θ. In the same way, $C^1_{t_1} C^1_{t_2} C^3_{t_3}$ can also be discovered as an MCSC, since $\frac{\left|C^1_{t_1} \cap C^1_{t_2}\right|}{\left|C^1_{t_1} \cup C^1_{t_2}\right|} = \frac{1}{3}$, $\frac{\left|C^1_{t_2} \cap C^3_{t_3}\right|}{\left|C^1_{t_2} \cup C^3_{t_3}\right|} = \frac{1}{2}$. $C^1_{t_1}$ and $C^1_{t_2}$ are different representations of general class C^1 at timestamps t_1 and t_2. After removing the timestamp subscripts in the sequence of snapshot clusters and the sequence data deduplication, we get two sequences of general classes, which are $C^1C^2C^3$ and C^1C^3. We have every reason to believe that $C^1C^2C^3$ and C^1C^3 are the two typical group moving behaviors in the scene during this period.

4 Semantic scene modeling using markov topic model

4.1 Markov topic model

A traffic video sample $\text{V} = \{v_1, v_2, \ldots, v_T\}$ can be divided into t-segment video clips, $1 \leq t \leq T$. In our method, a video is segmented into non-overlapping clips with equal frame length N_f, whose value is between 100 and 300, depending on the nature of the traffic scene. We treat a

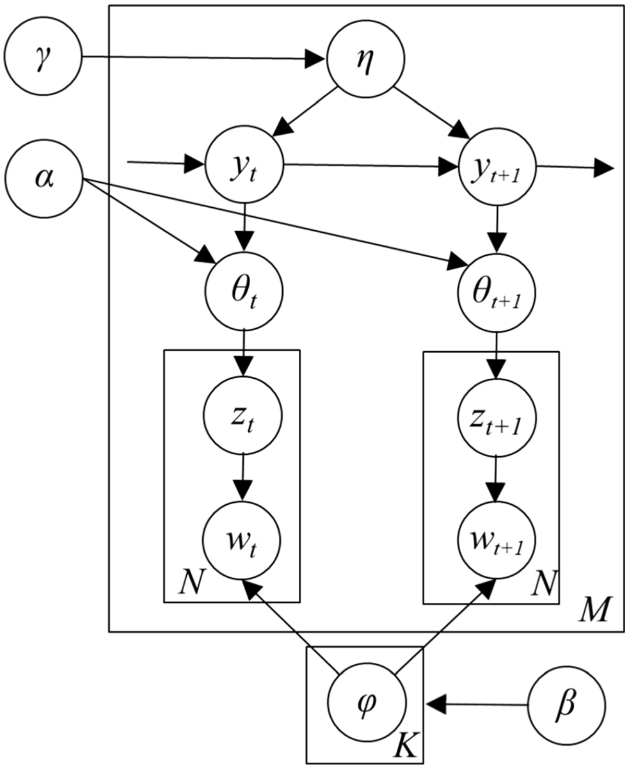


Fig. 2 Graphical representation of Markov topic model

video clip as a document, and the trajectory points of moving objects as motion words. Each video clip is mapped to a motion word bag and embodies a kind of behavior pattern. Finally, the video sample is represented by a behavior sequence (motion word package sequence). Because the topic model itself ignores the temporal relationship between words, it cannot model the temporal correlation between behaviors. We incorporate a hidden Markov chain to model these and propose an MTM to model a video clip sequence as a Markov chain of behaviors.

Figure 2 shows the graphical representation of an MTM. Let M traffic videos constitute a video set $\mathrm{D} = \{V_1, V_2, \ldots, V_M\}$, where each video $V_j (j = 1, 2, \ldots, M)$ is divided into t-segment video clips, and each video clip v_t shows a specific kind of behavior, y_t.

We model the behavior sequence $\{y_t\}_{t=1}^{T}$ as a Markov chain, and the behavior y_t evolves along the sequence direction of the video clip sequence according to a multinomial distribution,

$$\mathrm{p}(y_t | y_{t-1}, \eta) = \text{multinominal}\left(\eta_{y_{t-1}}\right). \tag{1}$$

The behavior pattern Y_j of video V_j is a behavior sequence composed of video clip behavior y_t,

$$Y_j = \{y_1, y_2, \ldots, y_t, \ldots, y_T\}. \tag{2}$$

All of the trajectory points of moving objects in each video clip v_t are encoded, and a video clip is represented as a word bag containing N motion words, where w_t is the set of motion words $w_{t,i}$ contained in v_t,

$$w_t = \left\{w_{t,1}, w_{t,2}, \ldots, w_{t,i}, \ldots, w_{t,N}\right\}. \tag{3}$$

Each video clip is represented by a random mixture distribution over K latent topics (semantic regions). Each topic in video clip v_t is essentially a distribution of trajectory points that constitute the semantic regions in the scene. The mixed ratio θ_t of various topics is determined by the Dirichlet parameter α as

$$\mathrm{p}(\theta_t | y_t, \alpha) = \text{Dirichlet}(\alpha). \tag{4}$$

The mixed ratio θ_t is the multinomial distribution of topics in the video clip v_t, and the topic $z_{t,i}$ is determined by the parameter θ_t as

$$\mathrm{p}\left(z_{t,i} | \theta_t\right) = \text{multinominal}(\theta_t). \tag{5}$$

The motion word $w_{t,i}$ in video clip v_t is determined by φ_{z_t} as

$$\mathrm{p}\left(w_{t,i} | z_{t,i}, \varphi\right) = \text{multinominal}\left(\varphi_{z_t}\right). \tag{6}$$

The generative process of the MTM is described as follows:

1. For each latent topic $z \in 1, 2, \ldots, k, \ldots K$, draw discrete distributions φ_k of motion words from a Dirichlet prior β, and construct $K \times L$ dimensional matrix $\varphi \sim \mathit{Dirichlet}(\beta)$;
2. For each video V_j, $j \in 1, 2, \ldots, M$, draw discrete distributions η of behavior types from Dirichlet prior γ: $\eta \sim \text{Dirichlet}(\gamma)$;
3. Generate any video clip v_t in video V_j as follows:

   a. Sample the corresponding behavior type $y_t \sim \text{multinominal}\left(\eta_{y_{t-1}}\right)$;
   b. Draw discrete distribution θ_t of topics from a Dirichlet prior, α: $\theta_t \sim \text{Dirichlet}(\alpha)$;
   c. Generate motion word $w_{t,i}$ in video clip v_t as follows:

      i. Draw the corresponding topic $z_{t,i}$: $z_{t,i} \sim \text{multinominal}(\theta_t)$;
      ii. Draw motion word $w_{t,i}$ from $\mathrm{p}\left(w_{t,i} | z_{t,i}, \varphi\right)$.

The number L of basic motion words in the dictionary and the number K of topics are predetermined. The conditional probability $p\left(w_{t,i} | z_{t,i}, \varphi\right)$ is calculated by the matrix φ of dimension KL, where φ_{ij} is the generating probability of motion word w_j under the condition of the current known topic type z_i, i.e.,

$$\varphi_{ij} = p\left(w_j = 1 | z_i = 1\right). \tag{7}$$

Given α, γ, and β, the joint distribution of topic mixture parameter θ, motion word mixture parameter φ, behavior mixture parameter η, behavior type y_t, topics $z_t = \left\{z_{t,i}\right\}$, and motion words $w_t = \left\{w_{t,i}\right\}$ is

$$\begin{aligned} &\mathrm{p}\left(\{w_t, z_t, y_t\}_1^T, \eta, \theta, \varphi | \alpha, \beta, \gamma\right) \\ &= \mathrm{p}(\eta|\gamma)\mathrm{p}(\varphi|\beta) \prod_{t=1}^{T} \left(\prod_{i=1}^{N_j} \mathrm{p}\left(z_{t,i} | \theta_t\right) \mathrm{p}\left(w_{t,i} | z_{t,i}, \varphi\right) \right) \\ &\quad \mathrm{p}(y_t | y_{t-1}, \eta)\mathrm{p}(\theta_t | \alpha, y_t), \end{aligned} \tag{8}$$

where N_j is the number of motion words in video clip v_t.

4.2 Model learning

We adopt Gibbs sampling based on the Markov chain Monte Carlo method to estimate the parameters of MTM. Due to the Dirichlet-multinomial conjugate structures of the model, the parameters $\{\eta, \theta, \varphi\}$ can be automatically eliminated during the Gibbs sampling procedure.

The parameters θ and φ are eliminated from the conditional probability, and the Gibbs sampling conditional probability for topic is obtained as

$$\mathrm{p}\left(z_{t,i} | z_{t,\neg i}, y, w\right) \propto \frac{n_{z_t}^{w} + \beta}{\sum_w n_{z_t}^{w} + L\beta} \frac{n_{\mathrm{y}}^{z} + \alpha}{\sum_z n_{\mathrm{y}}^{z} + K\alpha}, \tag{9}$$

where $z_{t,\neg i}$ denotes all z variables except $z_{t,i}$, $n_{z_t}^{w}$ is the number of basic motion words in the dictionary assigned to topic z_t, $\sum_{w} n_{z_t}^{w}$ is the total number of motion words assigned to topic z_t, n_{y}^{z} is the number of topic types assigned to behavior type y, $\sum_{z} n_{\mathrm{y}}^{z}$ is the total number of topics assigned to behavior type y, L is the number of basic motion words in the dictionary, and K is the number of topic types.

The parameters η and θ are eliminated from the conditional probability, and the Gibbs sampling conditional probability for behavior type is obtained as

$$\mathrm{p}(y_t|y_{\neg t}, z, w) \propto \frac{\prod_z \Gamma\left(\alpha + n_{\mathrm{y}_t}^{z}\right)\Gamma\left(K\alpha + \sum_z n_{\neg \mathrm{y}_t}^{z}\right)}{\prod_z \Gamma\left(\alpha + n_{\neg \mathrm{y}_t}^{z}\right)\Gamma\left(K\alpha + \sum_z n_{\mathrm{y}_t}^{z}\right)} \frac{\left(n_{y_t,y_{t+1}} + \gamma\right)\left(n_{y_t,y_{t+1}} + I(y_{t-1} = y_t)I(y_t = y_{t+1}) + \gamma\right)}{\sum_{y_{t+1}} n_{y_t,y_{t+1}} + I(y_{t-1} = y_t) + M\gamma}, \quad (10)$$

where $y_{\neg t}$ denotes all y variables except y_t; $n_{y_t}^{z}$ is the number of topic types assigned to behavior types y_t; $n_{\neg \mathrm{y}_t}^{z}$ is the number of topic types not assigned to behavior types y_t; $\sum_{z} n_{\mathrm{y}_t}^{z}$ is the total number of topics assigned to behavior type y_t; $\sum_{z} n_{\neg \mathrm{y}_t}^{z}$ is the total number of topics not assigned to behavior type y_t; $n_{y_t,y_{t+1}}$ is the count of behavior type y_{t+1} following behavior type y_t; $\sum_{y_{t+1}} n_{y_t,y_{t+1}}$ is the total count of behavior type y_t, followed by other behavior types; M is the number of behavior types; and I is the identity function, which returns the value 1 if its argument is true.

After sampling converges, parameters can be estimated as

$$\hat{\theta} = \frac{n_{\mathrm{y}}^{z} + \alpha}{\sum_z n_{\mathrm{y}}^{z} + K\alpha}, \quad (11)$$

$$\hat{\varphi} = \frac{n_{z}^{w} + \beta}{\sum_w n_{z}^{w} + L\beta}, \quad (12)$$

$$\hat{\eta} = \frac{n_{y_t,y_{t+1}} + \gamma}{\sum_{y_{t+1}} n_{y_t,y_{t+1}} + M\gamma}. \quad (13)$$

5 Discovery algorithms of MCSC patterns

By learning from MTM, the trajectory points of moving objects (co-occurring motion words) in video clips (documents) can be clustered into semantic regions (topics). Then we get the set of snapshot clusters $C_{t_i} = \left\{C_{t_i}^{1}, C_{t_i}^{2}, \ldots, C_{t_i}^{K}\right\}$ at timestamp t_i according to the description of the mapping process in Sect. 3, where cluster $C_{t_i}^{j} = \left\{o_{j1}, o_{j2}, \ldots, o_{jv}\right\}(1 \le j \le K, 1 \le v \le m)$ is one of the snapshot clusters at timestamp t_i, K is the number of semantic regions (topics), and m is the total number of moving objects.

The MCSCs can be discovered by using the candidate MCSC recognition algorithm to construct lists of candidate MCSCs and using the screening algorithm to distinguish MCSCs in the candidate lists. We give the following definitions.

Definition 2 *(Key Set of Moving Objects):* Let $C_{t_{i-1}}^{p}$ and $C_{t_i}^{q}$ be the snapshot clusters at timestamp t_{i-1} and t_i, respectively, and $S = \{o_{i1}, o_{i2}, \ldots, o_{iu}\}$ a set of moving objects at timestamp t_i. S is said to be a **key set of moving objects** if:

1) $\frac{\left|C_{t_{i-1}}^{p} \cap C_{t_i}^{q}\right|}{\left|C_{t_{i-1}}^{p} \cup C_{t_i}^{q}\right|} \ge \theta(0 < \theta \le 1)$: $C_{t_{i-1}}^{p} \cap C_{t_i}^{q}$ is an MCSC;

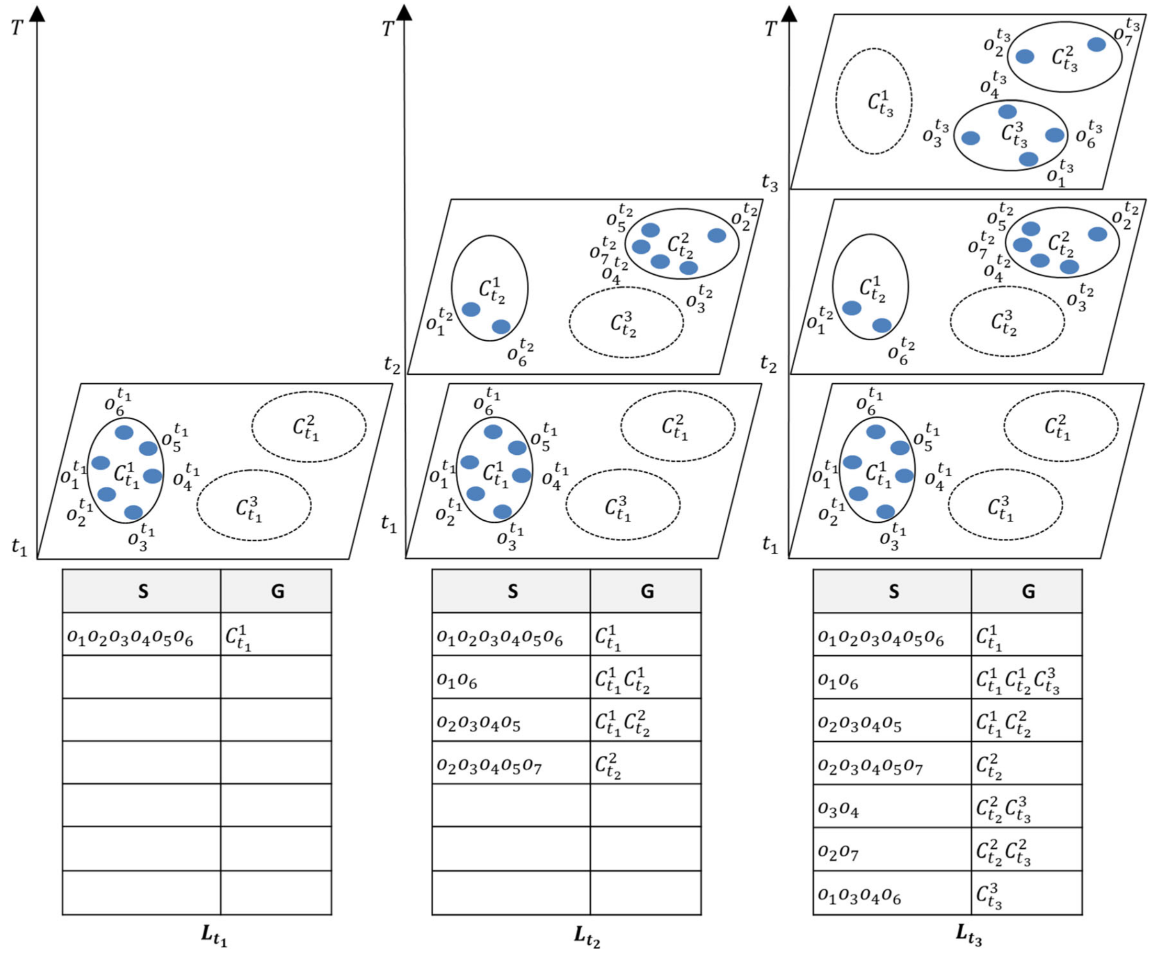
2) $S = C_{t_{i-1}}^{p} \cap C_{t_i}^{q}$: these moving objects are the decisive factors for MCSC;

3) $|S| \ge minS$: there are at least $minS$ objects in set S.

Definition 3 **(Candidate MCSC):** Let S be a key set of moving objects. S, G is a candidate MCSC, where S is the intersection of the set of moving objects contained in snapshot cluster $C_{t_{i-1}}^{p}$ and $C_{t_i}^{q}$, and $G = C_{t_{i-1}}^{p} C_{t_i}^{q}$.

A queue L of candidate MCSCs is called the list of candidate MCSCs. At timestamp t_i, all current candidate MCSCs are stored in list L.

The process of identifying MCSCs is to construct and update the lists of candidate MCSCs at each timestamp, step by step. Figure 3 shows a running example of the candidate MCSC recognition algorithm (CMCR). There are seven objects and three timestamps ($O = \{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}$, $T = \{t_1, t_2, t_3\}$), and $\mathrm{o}_j^{t_1}$, $\mathrm{o}_j^{t_2}$ and $\mathrm{o}_j^{t_3}$ are the representations of moving object o_j at different timestamps. We set $\theta = \frac{1}{4}$ and $minS = 2$ in this example. At each timestamp, the key set of moving objects is constructed and updated in list L together with the corresponding sets of snapshot clusters. The lists corresponding to timestamps t_1, t_2, and t_3 are L_{t_1}, L_{t_2}, and L_{t_3}, respectively.


T
T
T
t_1
t_1
t_1
t_2
t_2
t_3
$C_{t_1}^1$
$C_{t_1}^2$
$C_{t_1}^3$
$C_{t_2}^1$
$C_{t_2}^2$
$C_{t_2}^3$
$C_{t_3}^1$
$C_{t_3}^2$
$C_{t_3}^3$
$o_1^{t_1}$ $o_2^{t_1}$ $o_3^{t_1}$ $o_4^{t_1}$ $o_5^{t_1}$ $o_6^{t_1}$
$o_1^{t_2}$ $o_2^{t_2}$ $o_3^{t_2}$ $o_4^{t_2}$ $o_5^{t_2}$ $o_6^{t_2}$ $o_7^{t_2}$
$o_1^{t_3}$ $o_2^{t_3}$ $o_3^{t_3}$ $o_4^{t_3}$ $o_6^{t_3}$ $o_7^{t_3}$

S	G
$o_1o_2o_3o_4o_5o_6$	$C_{t_1}^1$

L_{t_1}

S	G
$o_1o_2o_3o_4o_5o_6$	$C_{t_1}^1$
o_1o_6	$C_{t_1}^1C_{t_2}^1$
$o_2o_3o_4o_5$	$C_{t_1}^1C_{t_2}^2$
$o_2o_3o_4o_5o_7$	$C_{t_2}^2$

L_{t_2}

S	G
$o_1o_2o_3o_4o_5o_6$	$C_{t_1}^1$
o_1o_6	$C_{t_1}^1C_{t_2}^1C_{t_3}^3$
$o_2o_3o_4o_5$	$C_{t_1}^1C_{t_2}^2$
$o_2o_3o_4o_5o_7$	$C_{t_2}^2$
o_3o_4	$C_{t_2}^2C_{t_3}^3$
o_2o_7	$C_{t_2}^2C_{t_3}^2$
$o_1o_3o_4o_6$	$C_{t_3}^3$

L_{t_3}

Fig. 3 Example of CMCR algorithm

5.1 Update and insertion rules for candidates

According to the above definitions, we introduce the following lemma.

Lemma 1 *Let $C_{t_{i-1}}^pC_{t_i}^q$, and $C_{t_i}^qC_{t_{i+1}}^r$ be MCSCs. $C_{t_{i-1}}^pC_{t_i}^qC_{t_{i+1}}^r$ is an MCSC.*

Proof Since $C_{t_{i-1}}^pC_{t_i}^q$ and $C_{t_i}^qC_{t_{i+1}}^r$ are MCSCs, we have $\frac{\left|C_{t_{i-1}}^p\cap C_{t_i}^q\right|}{\left|C_{t_{i-1}}^p\cup C_{t_i}^q\right|}\geq\theta$ and $\frac{\left|C_{t_i}^q\cap C_{t_{i+1}}^r\right|}{\left|C_{t_i}^q\cup C_{t_{i+1}}^r\right|}\geq\theta$. Therefore, $C_{t_{i-1}}^pC_{t_i}^qC_{t_{i+1}}^r$ is an MCSC.

We give the following update and insertion rules.

Rule 1 (Update): For any candidate item $u=C_{t_{i-1}}^p,G$ in list L, if $C_{t_{i-1}}^p\subseteq C_{t_i}^q$ and $\frac{\left|C_{t_{i-1}}^p\cap C_{t_i}^q\right|}{\left|C_{t_{i-1}}^p\cup C_{t_i}^q\right|}\geq\theta$, then candidate item u is updated to $C_{t_{i-1}}^p,G\cup\left\{C_{t_i}^q\right\}$.

In Fig. 3, $\{o_1,o_6\},C_{t_1}^1C_{t_2}^1$ is a candidate in list L_{t_2}, where $C_{t_2}^1=\{o_1,o_6\}$ and $C_{t_3}^3=\{o_1,o_3,o_4,o_6\}$. The candidate item $\{o_1,o_6\},C_{t_1}^1C_{t_2}^1$ is updated to $\{o_1,o_6\},C_{t_1}^1C_{t_2}^1C_{t_3}^3$, because $C_{t_2}^1\subseteq C_{t_3}^3$ and $\frac{\left|C_{t_2}^1\cap C_{t_3}^3\right|}{\left|C_{t_2}^1\cup C_{t_3}^3\right|}=\frac{1}{2}\geq\theta$.

In fact, candidate item u is extended by snapshot cluster $C_{t_3}^3$ in L_{t_3} to form a valid MCSC. It should be noted that $C_{t_1}^1$ and $C_{t_2}^1$ are different representations of the learned semantic region C^1 at timestamps t_1 and t_2, respectively.

Rule 2: For any candidate item $u=C_{t_{i-1}}^p,G$ in list L, if $C_{t_{i-1}}^p\subsetneq C_{t_i}^q$ and $\frac{\left|C_{t_{i-1}}^p\cap C_{t_i}^q\right|}{\left|C_{t_{i-1}}^p\cup C_{t_i}^q\right|}\geq\theta$, then a new candidate item $C_{t_{i-1}}^p\cap C_{t_i}^q,G\cup\left\{C_{t_i}^q\right\}$ is created and inserted in list L.

In Fig. 3, $\{o_1,o_6\},C_{t_1}^1C_{t_2}^1$, $\{o_2,o_3,o_4,o_5\},C_{t_1}^1C_{t_2}^2$, $\{o_3,o_4\},C_{t_2}^2C_{t_3}^3$, and $\{o_2,o_7\},C_{t_2}^2C_{t_3}^2$ are all candidate items created under rule 2.

Rule 3: For any snapshot cluster $C_{t_i}^q$ at timestamp t_i, if $C_{t_i}^q\subsetneq C_{t_{i-1}}^p$, then a new candidate item $C_{t_i}^q,\left\{C_{t_i}^q\right\}$ is created and inserted in list L.

Obviously, $\{o_2,o_3,o_4,o_5,o_7\},C_{t_2}^2$ and $\{o_1,o_3,o_4,o_6\},C_{t_3}^3$ are two typical examples.

$\{o_1, o_2, o_3, o_4, o_5, o_6\}$, $C^1_{t_1}$ is the initial candidate item that should be inserted in list L when initialized.

5.2 Candidate MCSC recognition algorithm

The above rules construct the list L of candidate MCSCs and the set of added candidate items at current timestamp t_i. The candidate MCSC recognition algorithm maintains candidate list L by updating and inserting candidate items in list L according to rules 1–3 at each timestamp.

Algorithm 1 presents the pseudocode of CMCR.

Algorithm 1: Candidate MCSC Recognition Algorithm
Input: candidate list L at t_{i-1} and t_i, sets $C_{t_{i-1}}$ and C_{t_i} of snapshot clusters

Output: candidate list L at t_i, set $\vartriangle L_{t_i}$ of added candidate items at t_i

```
1:  $L_{t_1} \leftarrow \langle C^j_{t_1}, \{C^j_{t_1}\}\rangle$;// insert nonempty snapshot clusters at $t_1$ as initial candidate items
2:  $\vartriangle L_{t_1} \leftarrow \langle C^j_{t_1}, \{C^j_{t_1}\}\rangle$;
3:  for i ← 2 to n do // for each timestamp
4:    for each candidate item $u \leftarrow \langle C^p_{t_{i-1}}, G\rangle$ do
5:      if $\frac{|C^p_{t_{i-1}} \cap C^q_{t_i}|}{|C^p_{t_{i-1}} \cup C^q_{t_i}|} \geq \theta$ then
6:        if $C^p_{t_{i-1}} \subseteq C^q_{t_i}$ then
7:          $u.G \leftarrow u.G \cup \{C^q_{t_i}\}$; // update
8:        else
9:          new $v \leftarrow \langle C^p_{t_{i-1}} \cap C^q_{t_i}, G \cup \{C^q_{t_i}\}\rangle$; // create a new candidate item
10:         insertList $(v, \vartriangle L_{t_i})$; // insert
11:   for each $C^q_{t_i}$ do
12:     if $C^q_{t_i} \nsubseteq C^p_{t_{i-1}}$ then
13:       $v \leftarrow \langle C^q_{t_i}, \{C^q_{t_i}\}\rangle$;
15:       insertList $(v, \vartriangle L_{t_i})$;
16:   $L \leftarrow L + \vartriangle L_{t_i}$;
17:   output $\vartriangle L_{t_i}$ and $L$;
```

In the first step, at timestamp t_1, we generate initial candidate items utilizing the nonempty snapshot clusters in snapshot S_1, and insert them in $\vartriangle L_{t_1}$ and L (lines 1–2). At each subsequent timestamp, snapshots S_i and S_{i-1} are examined to update and insert candidate items (lines 4–10). When scanning through snapshot S_i, $\triangle L_{t_i}$ includes candidate items containing snapshot clusters in S_i (lines 11–15). The set $\triangle L$ of added candidate items at each timestamp is used to extend candidate MCSCs to closed candidates in the screening algorithm.

5.3 Screening algorithm

Our goal is to discover the complete set of MCSCs from the candidates, i.e., the MCSCs with the largest continuous timestamps. To avoid mining redundant MCSCs, we define a closed candidate.

Definition 4 (Closed Candidate):

For a candidate MCSC u in list L, if there does not exist another candidate v such that $u.G \subseteq v.G$, then u is a closed candidate.

We use the example in **Fig. 2** to intuitively explain our methods. It is easy to see that $\{o_2, o_3, o_4, o_5\}, \left\{C^1_{t_1}, C^2_{t_2}\right\}$ is a candidate MCSC at consecutive timestamps t_1 and t_2, and $\{o_3, o_4\}, \left\{C^2_{t_2}, C^3_{t_3}\right\}$ is a candidate MCSC at timestamps t_2 and t_3. It is obviously redundant to output $\{o_2, o_3, o_4, o_5\}, \left\{C^1_{t_1}, C^2_{t_2}\right\}$ as an MCSC, since it can be extended to form another MCSC, $\{o_3, o_4\}, \left\{C^1_{t_1}, C^2_{t_2}, C^3_{t_3}\right\}$, which is a closed candidate.

Based on the above output set $\triangle L$ at each timestamp, the screening algorithm can retrieve all MCSCs based on the construction of closed candidates. This is described as Algorithm 2.

Algorithm 2: Screening Algorithm
Input: $\triangle L$ at each timestamp
Output: set Φ of MCSCs

```
1:  $\Phi \leftarrow \emptyset$;
2:  $\triangle L_{t_1} \leftarrow \langle C^j_{t_1}, \{C^j_{t_1}\}\rangle$;
3:  for i ← 2 to n do // for each timestamp
4:    for each candidate item $u \in \triangle L_{t_{i-1}}$ do
5:      $u.extended \leftarrow false$;
6:      for each candidate item $v \in \triangle L_{t_i}$ do // extend candidate MCSCs to closed candidates
7:        if $u.G.tail = v.G.head$ then
8:          $g \leftarrow$ desplic $(u.G,\ v.G)$; // deduplicate and splice MCSCs
9:          insertList $(\langle v.S\ g\rangle,\ \triangle L_{t_i})$; // to be used at next iteration
10:          $u.extended \leftarrow true$;
11:      if (not $u.extended$) then
12:        regsub $(u.G)$; // deduplicate in sequence of snapshot clusters (MCSCs)
13:        insert $(u.G,\ \Phi)$;
14:        output $\Phi$;
```

When examining $\triangle L_{t_{i-1}}$, $\triangle L_{t_{i-1}}$ includes the added candidate items at timestamp t_{i-1}. Each candidate item $u \in \triangle L_{t_{i-1}}$ is checked to verify whether u can be extended by any $v \in \triangle L_{t_i}$ to form a valid MCSC (lines 6–10). The candidate items in $\triangle L_{t_{i-1}}$ that were not extended at t_i are output. $\triangle L_{t_i}$ to be used at the next iteration consists of candidate items in $\triangle L_{t_{i-1}}$ that were extended at t_i. In this manner, the screening algorithm misses no closed candidates and outputs no redundant items.

Note that the original snapshot cluster $C^1_{t_1}$ splits into snapshot clusters $C^1_{t_2}$ and $C^2_{t_2}$ at timestamp t_2. Since $\frac{\left|C^1_{t_1}\cap C^1_{t_2}\right|}{\left|C^1_{t_1}\cup C^1_{t_2}\right|} = \frac{1}{3} \geq \theta$ and $\frac{\left|C^1_{t_1}\cap C^2_{t_2}\right|}{\left|C^1_{t_1}\cup C^2_{t_2}\right|} = \frac{4}{7} \geq \theta$, they satisfy the criterion of Definition 1. Therefore, $\{o_1, o_6\}, C^1_{t_1}C^1_{t_2}$ and $\{o_2, o_3, o_4, o_5\}, C^1_{t_1}C^2_{t_2}$ are legal candidate items in list L. This is called **traffic divergence**, and it captures the behavior of a group of objects that diverge into multiple semantic regions. This can be semantically described as a group of moving objects $\{o_1, o_2, o_3, o_4, o_5, o_6\}$ located in region 1 at timestamp t_1, and partial moving objects $\{o_2, o_3, o_4, o_5\}$ diverting to region 2 at timestamp t_2. In the case of symmetry, a similar phenomenon is observed, i.e., two snapshot clusters merge into one snapshot cluster at the next timestamp, which is called **traffic convergence**.

6 Experimental evaluation

6.1 Datasets

We evaluated the effectiveness of the proposed framework of MCSC discovery for group behavior pattern mining using QMUL junction and roundabout datasets. The two datasets were collected from real-world public road surveillance scenarios with large numbers of moving objects exhibiting typical group behavior patterns.

Junction Dataset This dataset contains 60 min of 25 fps video of a busy urban road junction with a frame size of 360 × 288 pixels. The scene contains many objects moving at different regions and exhibiting complex behaviors. The traffic lights govern the behaviors of moving objects, which form four types of traffic flow patterns in a certain temporal order, as shown in Fig. 4.

Roundabout Dataset This dataset contains about 60 min of 25 fps video of a traffic roundabout with a frame size of 360 × 288 pixels. Similar to the urban road junction, moving objects show complex behaviors. Three types of traffic flow patterns are controlled by traffic lights in a certain temporal order, as shown in Fig. 5.

6.2 Semantic region segmentation

Experiments were performed to decompose each scene into semantic regions on the QMUL junction and roundabout datasets. For the junction, 33 non-overlapping sub-videos were obtained after segmenting the video according to the temporal order of traffic flow. Each sub-video contains all traffic flow patterns in the scene, and they constitute video set D. The roundabout video dataset was similarly cut into 66 non-overlapping sub-videos, constituting video set D. Each sub-video was segmented into non-overlapping clips with an equal length of 100 frames. Each video clip was considered a document, and the trajectory points of moving objects within video clips were considered motion words in documents. To build the motion dictionary, the 360 × 288 surveillance scene was quantized into 40 × 32 cells of size 9 × 9, and the moving directions were quantized into four directions perpendicular to each other. So, the size of the dictionary was 40 × 32 × 4.

We set the model parameters $\alpha = K/50$, $\beta = 0.01$, $\gamma = 1$, and segmented each scene into semantic regions. K is the number of topics (semantic regions). Our proposed method decomposed the junction and roundabout scenes into nine regions, as shown in Figs. 6a and 7a. For comparison, in Figs. 6b and 7b, the junction and roundabout scenes were segmented into six and nine regions, respectively, using the modified spectral clustering algorithm of Li et al. [34], which represents the state of the art of semantic scene segmentation. Li et al. treated the semantic region segmentation problem as an image segmentation problem, except that they represented each pixel location of a scene by pixel-wise feature vectors instead of RGB and texture feature values. The modified spectral clustering algorithm was employed to segment the scene into different non-overlapping semantic regions. As shown in Figs. 6b and 7b, those areas not belonging to traffic lanes and waiting zones were also divided into semantic regions. From this point of view, the result of region segmentation

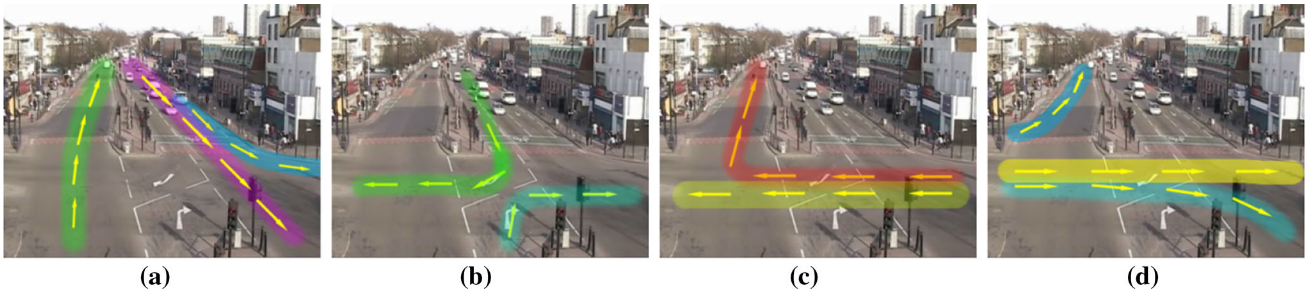

(a) (b) (c) (d)

Fig. 4 **a–d**: traffic flow patterns A–D, respectively, at urban road junction

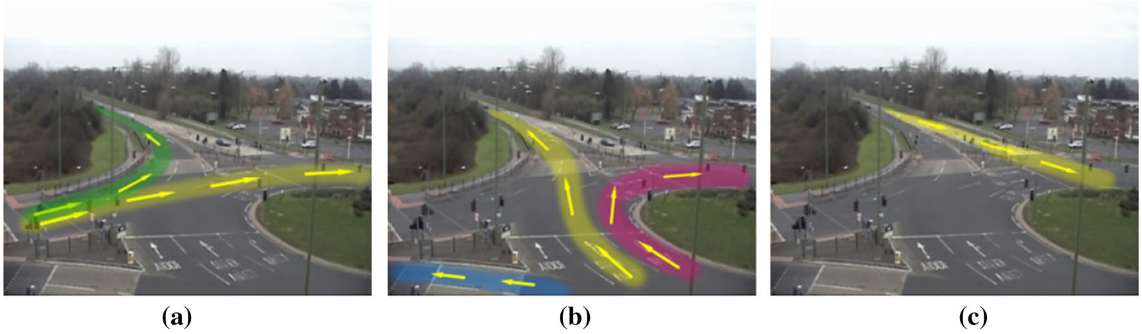
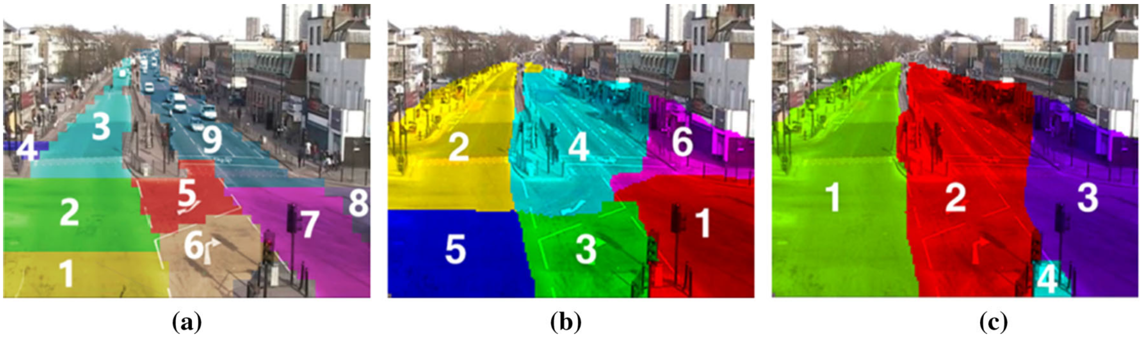
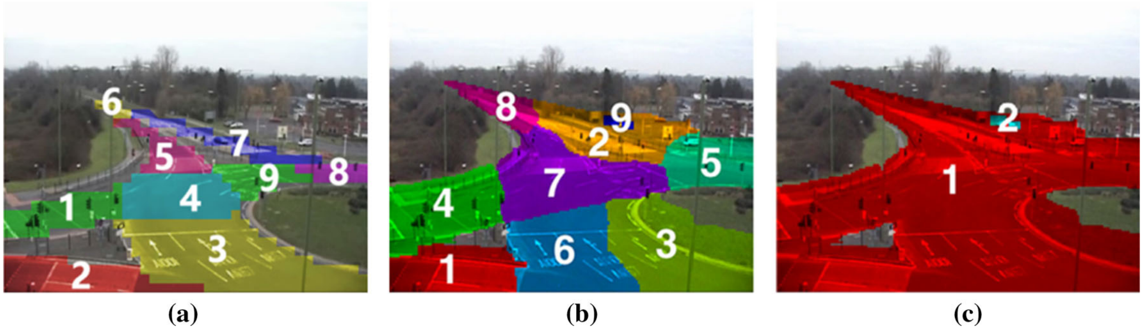

(a) (b) (c)

Fig. 5 **a**–**c**: traffic flow patterns E–G, respectively, at traffic roundabout



(a) (b) (c)

Fig. 6 Semantic region segmentation for junction dataset: **a** proposed method—9 regions; **b** spectral clustering algorithm—6 regions; **c** Zelnik–Perona method—4 regions



(a) (b) (c)

Fig. 7 Semantic region segmentation for roundabout dataset: **a** proposed method—9 regions; **b** spectral clustering algorithm—9 regions; **c** Zelnik–Perona method—2 regions

is somewhat rough. This segmentation result has been useful enough for learning behavior spatial context, but is unsuitable for our MCSC discovery scenarios. Other than that, both methods display meaningful distributions of semantic regions, which almost correspond to various traffic lanes and waiting zones. Moving objects passing through the same combination of semantic regions tend to follow a similar behavior pattern. Thus, it can be seen that inappropriate segmentation of semantic regions makes it more difficult to find meaningful group behavior patterns. For example, Figs. 6c and 7c show the segmented semantic regions using the original Zelnik–Perona (ZP) method [33] at an urban road junction and traffic roundabout, from which it is evident that, no matter $C^1C^2C^3$ and $C^3C^2C^1$ at the urban road junction, or C^1C^2 and C^2C^1 at the traffic roundabout, these are not meaningful group moving behaviors, which cannot reflect meaningful traffic flow patterns in real scenarios.

6.3 Traffic group behavior pattern mining evaluation

We evaluate the identified quality of traffic divergence, which is the typical traffic group behavior pattern, as obtained by MCSC discovery. With the learned semantic regions, the set of snapshot clusters at each timestamp is obtained by mapping moving objects to semantic regions (see Sect. 3). In the real world, streaming trajectory data generated by moving objects may arrive in different semantic regions of the monitoring scenarios at different timestamps. With rigorous constraints on time, inappropriate setting of timestamps makes it more difficult to discover meaningful MCSCs.

For all of the datasets of the experiments, considering the lifetime of moving objects in these scenarios, the interval between consecutive timestamps was set at one second, i.e., the movement data of moving objects were output each second to produce snapshot clusters. To retrieve all of the MCSCs, the candidate list L was reported at each timestamp in the form of S, G in the MCSC decision phase. G is considered to discover MCSCs by constructing closed candidates. The remaining S is the intersection of the set of moving objects contained in two adjacent snapshot clusters. This can be applied to extract more information, such as to identify the key moving objects from the process of generating MCSCs and describing their semantic behaviors, which is beyond the scope of this paper.

We generated ground truth by manually and exhaustively labeling all traffic divergence in the video timestamps from both datasets. The outputs of the different approaches were compared with the ground truth and evaluated by *Precision* and *Recall*, which are proportions of correct discoveries based on all retrieved results and the ground truth, respectively.

Baselines: We compared the proposed MCSC discovery approach with three state-of-the-art baselines: 1) the swarm (SW) pattern [12], which captures groups whose members move together in clusters of arbitrary shape for certain (possibly non-consecutive) time intervals; 2) the traveling companion (TC) pattern [13], which requires that group members are close together for certain consecutive snapshots; 3) and the moving cluster (MC) [15], which allows members to leave or join a group during its life cycle, where the proportion of common members in any two consecutive snapshot clusters does not fall below a threshold.

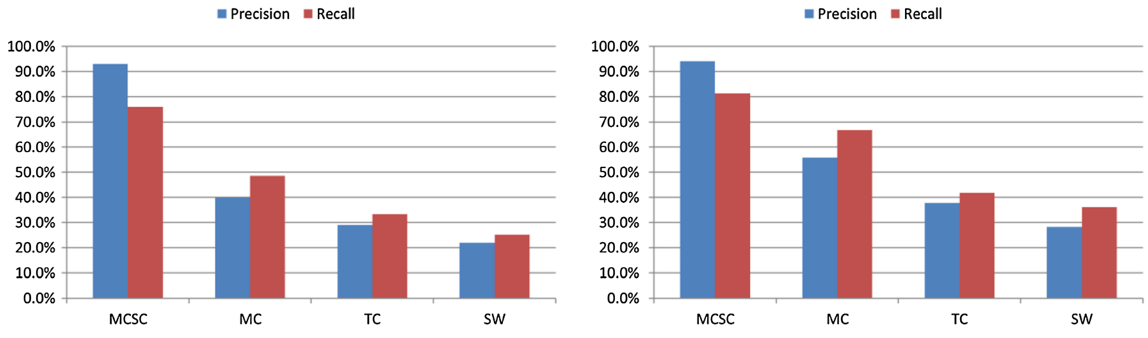
Parameter settings: Experimental parameters were set based on observations on the two datasets, and were selected intuitively according to the nature of the traffic scene. The sample points query operation in radius range widely used in DBSCAN [19] is supported by dividing the surveillance scene into 40 × 32 cells of size 9 × 9, i.e., 42 rows and 30 columns. Objects distributed on both sides of the traffic lane in parallel were required to be in a cluster. DBSCAN with $MinPts = 2$ and $Eps = 6$ was applied to generate clusters for SW, TC, and MC patterns at each timestamp. We set default size thresholds of $\min_o = 2$ (number of objects) and $\min_t = 2$ (consecutive timestamps, i.e., half of the shortest time span of traffic flows) for both datasets. For discovery of MC and MCSC patterns, we considered $\theta = \frac{1}{4}$ and separately for MCSC patterns, we set $\min S = 1$, which are restricted by scene structures.

Figure 8a and b plot the precision and recall of different approaches on the junction and roundabout datasets, according to the default setting. As indicated in Fig. 8a, MCSC, MC, TC, and SW achieved precision of 93.0%, 40.0%, 29.0%, and 21.9%, respectively, on the junction dataset, with corresponding recall of 76.0%, 48.5%, 33.3%, and 25.1%. On the roundabout dataset, Fig. 8b indicates precision for MCSC, MC, TC, and SW of 94.0%, 55.8%, 37.8%, and 28.2%, respectively, with corresponding recall of 81.3%, 66.7%, 41.8%, and 36.1%. As shown in figures, our proposed approach gains the preferable performance on both traffic scenarios. SW, TC, and MC utilized the DBSCAN algorithm to generate snapshot clusters at each timestamp, which cannot well reflect datasets with changeable data density because of the adopted global density parameter. When the density of the dataset is not uniform and the distance between clusters is very different, the quality of clustering is poor. In the two datasets, under different traffic flow conditions, the density of vehicles varied greatly. In extreme cases, all high-density moving objects were in one group, so it was difficult to distinguish snapshot clusters. In the case of sparse moving objects, objects with different motion behaviors could be grouped into one class, resulting in more false positives.

In traffic monitoring scenarios with entering or exiting moving objects, MC is more suitable than SW and TC because MC groups satisfy the overlap threshold between snapshot clusters distributed on consecutive timestamps, while the latter are the results of the intersection of the (continuous or with an allowed time-gap) snapshot clusters. SW generates more false positives that degrade performance. In addition, the performance of SW, TC, and MC on the roundabout dataset was significantly better than on the junction dataset. This is because DBSCAN does not consider the moving directions of objects, which leads to more false positives when the spatial positions of moving objects are adjacent and the moving directions are opposite. Traffic flow pattern B (see Fig. 4b) at an urban road junction is a typical scene of false positives.

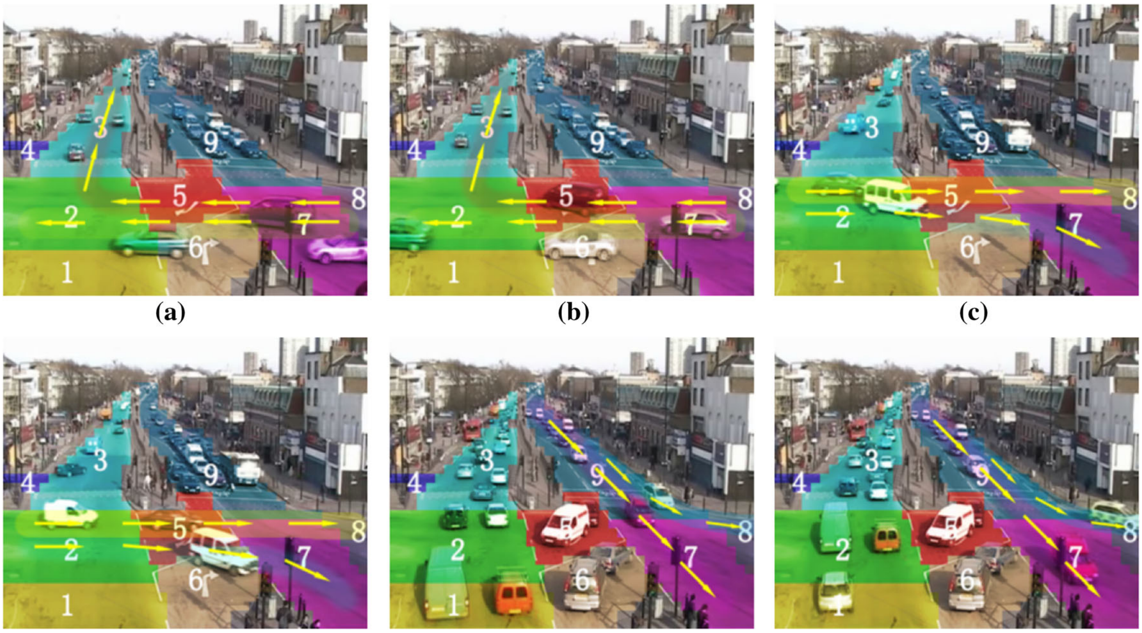
It is also interesting that the high-frequency traffic differences identified from the junction and roundabout datasets can reflect their typical traffic flow patterns, as shown in Figs. 9 and 10.

To further investigate the influences of parameters K and θ on the quality of traffic divergence identification, we used the junction dataset from the previous experiment and varied K from 6 to 12, and θ from 1/4 to 1/2.



(a) (b)

Fig. 8 Precision and recall of different approaches on: **a** junction dataset; and **b** roundabout dataset



(a) (b) (c)

(d) (e) (f)

Fig. 9 Traffic divergence identified at urban road junction corresponding to: **a**–**b** traffic flow pattern C; **c**–**d** traffic flow pattern D; **e**–**f** traffic flow pattern A

As shown in Fig. 11, both the precision and recall increase when K increases in the range of 6–9, because semantic regions that better conform to the scene structure identify more traffic divergences with fewer false positives. When K increases in the range of 9–12, the precision increases and recall decreases because inappropriately segmented semantic regions eliminate traffic divergences that fail to satisfy the θ criterion, and the probability of error recognition is smaller. If K is set greater than 11, 100% precision can be achieved. However, the recall under this setting is extremely low. We also observe that the precision increases and recall decreases when θ increases, because when θ is large, the probability of discovering MCSCs at consecutive timestamps is small, and the probability of false positives also decreases.

Finally, we conducted experiments while changing the values of the size threshold $\min_o$ on the junction dataset. Figure 12 shows the precision and recall of MCSC, MC, TC, and SW. It should be noted that we considered the default setting of $\min S = 1$ for MCSC instead of changing the value because, in a real-world traffic scene, a large number of snapshot clusters generated by mapping moving objects to semantic regions can only contain one moving object due to the size of the semantic regions. In addition, we did not consider the influence of the duration threshold $\min_t$ because that only two different snapshot clusters at two consecutive timestamps, respectively, are needed to determine whether traffic divergence is possible. Therefore, we took the duration threshold $\min_t$ as the default setting. As shown in Fig. 12, the recall of MC, TC, and SW

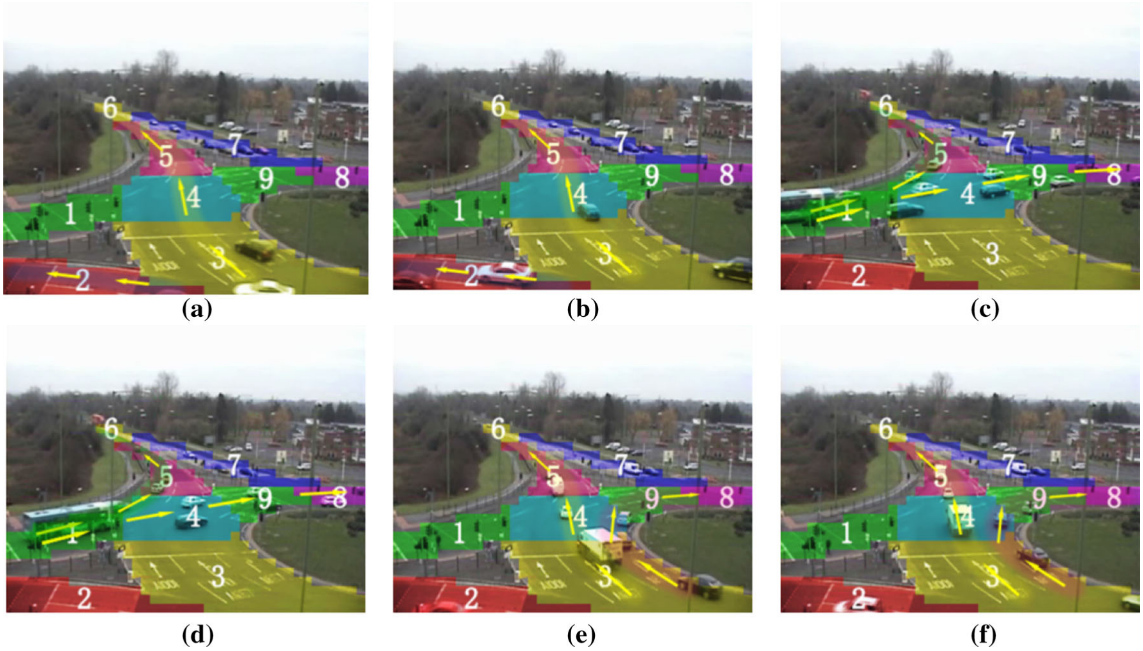


Fig. 10 Traffic divergence identified at traffic roundabout corresponding to: **a**–**b** traffic flow pattern F; **c**–**d** traffic flow pattern E; **e**–**f** traffic flow pattern F

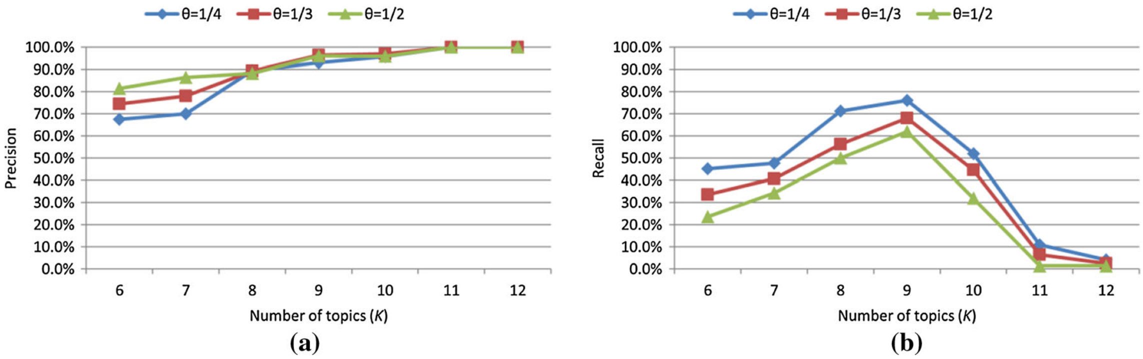


Fig. 11 Effectiveness of MCSC: **a** precision; **b** recall

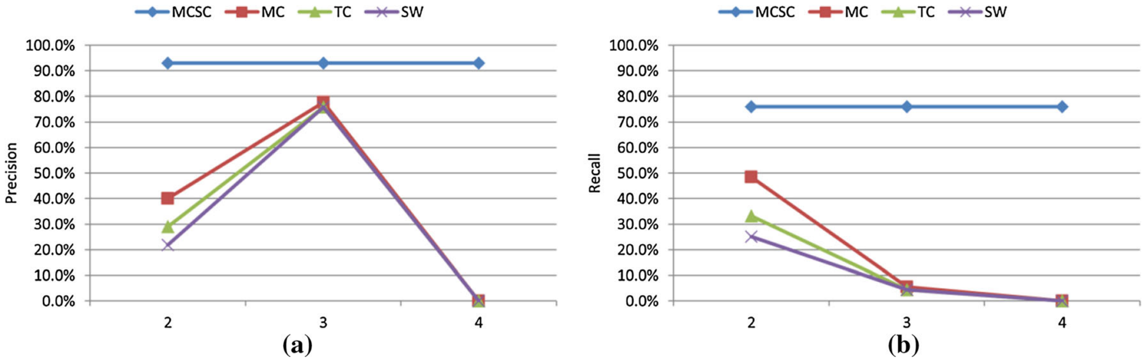


Fig. 12 Effectiveness: **a** precision and **b** recall of MCSC **vs** *min*$_o$

decreased as min$_o$ increased, because the snapshot clustering available for traffic divergence detection was reduced affected by semantic regions. When the duration threshold min$_o$ increased from 2 to 3, the precision of all approaches increased, because fewer snapshot clusters could pass a higher threshold with fewer false positives.

When $\min_o = 4$, precision and recall were both 0 because of no detected traffic divergence.

7 Conclusion

We introduced the challenging problem of MCSC discovery, which yields to group variability and behavioral regionality. Different from previous learned behavior clusters, the membership information of MCSC may change over time, and objects' motions show regional characteristics owing to scene structures. Since density-based clustering algorithms cannot reflect the movement of vehicles restricted by scene structures, we proposed a Markov topic model to segment semantic regions, which are viewed as subsets of a path that objects move along. At each timestamp, snapshot clusters are obtained by mapping moving objects to semantic regions, rather than through the conventional density-based clustering algorithms. Finally, a candidate MCSC recognition algorithm and screening algorithm, executed at each timestamp, incrementally identified and output MCSCs in real time. Extensive experiments on two real-world public road surveillance scenarios demonstrated the effectiveness of the proposed framework.

At present, the number of topics is mainly determined through domain knowledge or cross-validation. Our future work will focus on optimizing hyperparameters and intelligently training the number of topics. The coordination of multi-camera monitoring functions and the fusion of video data information are also a focus of further research.

Acknowledgements This research was supported in part by the National Natural Science Foundation of China under Grant No. 61672372, the Outstanding Science-Technology Innovation Team Program of Colleges and Universities in Jiangsu, the Natural Science Planning Project of Fujian Province under Grant 2020J01264, the Fujian Province Science and Technology Plan Project under Grant 2019J05123, the Young Teacher Education Research Project of Fujian under Grant JT180435, and the Natural Science Fund for Colleges and Universities in Jiangsu Province under Grant No. 19KJB520056.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Nguyen T, Armitage G (2008) A survey of techniques for Internet traffic classification using machine learning. IEEE Commun Surv Tutorials 10(4):56–76
2. Borges PVK, Conci N, Cavallaro A (2013) Video-based human behavior understanding: a survey. IEEE Trans Circuits Syst Video Technol 23(11):1993–2008
3. Hu W, Tan T, Wang L, Maybank S (2004) A survey on visual surveillance of object motion and behaviors. IEEE Trans Syst Man Cybern Part C Appl Rev 34(3):334–352
4. Veeraraghavan H, Papanikolopoulos NP (2010) Learning to recognize video-based spatiotemporal events. IEEE Trans Intell Transp Syst 10(4):628–638
5. Hu W, Li X, Tian G, Maybank S, Zhang Z (2013) An incremental DPMM-based method for trajectory clustering, modeling, and retrieval. IEEE Trans Pattern Anal Mach Intell 35(5):1051–1065
6. Eccles K, Fiedler R (2014) Automated enforcement for speeding and red light running. TR news: Transportation research
7. Kasper D, Weidl G, Dang T et al (2012) Object-oriented bayesian networks for detection of lane change maneuvers. IEEE Intell Transp Syst Mag 4(3):19–31
8. Jeung H, Yiu ML, Jensen CS (2011) Trajectory pattern mining, computing with spatial trajectories. Springer, New York, pp 143–177
9. Wang Y, Lim EP, Hwang SY (2003) On mining group patterns of mobile users. 14th International conference on database and expert systems applications (DEXA). Czech Republic, Prague, pp 287–296
10. Benkert M, Gudmundsson J, Hbner F, Wolle T (2008) Reporting flock patterns. Comput Geom 41(3):111–125
11. Jeung H, Yiu ML, Zhou X, Jensen C, Shen HT (2008) Discovery of convoys in trajectory databases. In: 34th international conference on very large data bases (VLDB). Auckland, New Zealand, pp 1068–1080
12. Li Z, Ding B, Han J, Kays R (2010) Swarm: mining relaxed temporal moving object clusters. In: 36th International Conference on Very Large Data Bases (VLDB). Singapore, pp 723–734
13. Tang LA, Zheng Y, Yuan J et al (2012) On discovery of traveling companions from streaming trajectories. IEEE 28th international conference on data engineering (ICDE). DC, USA, Washington, pp 186–197
14. Naserian E, Wang X, Xu X, Dong YN (2018) A framework of loose travelling companion discovery from human trajectories. IEEE Trans Mob Comput 17(11):2497–2511
15. Kalnis P, Mamoulis N, Bakiras S (2005) On discovering moving clusters in spatio-temporal data. In: 9th international conference on advances in spatial and temporal databases. Angra dos Reis, Brazil, pp 364–381
16. Zheng K, Zheng Y, Yuan NJ, Shang S (2013) On discovery of gathering patterns from trajectories. In: IEEE 29th international conference on data engineering (ICDE). Brisbane, QLD, AUS, pp 242–253
17. Zheng K, Zheng Y, Yuan NJ, Shang S, Zhou X (2014) Online Discovery of Gathering Patterns over Trajectories. IEEE Trans Knowl Data Eng 28(8):1974–1988
18. Aung HH, Tan KL (2010) Discovery of evolving convoys. In: 22nd international conference on scientific and statistical database management (SSDBM). Heidelberg, Germany, pp 196–213

19. Martin E, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: second international conference on knowledge discovery and data mining. Portland, Oregon, USA, pp 226–231
20. Ankerst M, Breunig M, Kriegel HP, Sander J (1999) OPTICS: Ordering points to identify the clustering structure. 1999 ACM SIGMOD international conference on management of data. Pennsylvania, USA, Philadelphia, pp 49–60
21. Hu W, Xie D, Fu Z, Zeng W, Maybank S (2007) Semantic-based surveillance video retrieval. IEEE Trans Image Process 16(4):1168–1181
22. Ng AY, Jordan MI, Weiss Y (2001) On Spectral Clustering: Analysis and an algorithm. In: 15th Annual Conference on Neural Information Processing Systems (NIPS). Vancouver, Canada, pp 849–856
23. Hu W, Xiao X, Fu Z, Xie D, Maybank S (2006) A system for learning statistical motion patterns. IEEE Trans Pattern Anal Mach Intell 28(9):1450–1464
24. Buzan D, Sclaroff S, Kollios G (2004) Extraction and clustering of motion trajectories in video. In: 17th international conference on pattern recognition (ICPR). Cambridge, UK, pp 521–524
25. Bashir FI, Khokhar AA, Schonfeld D (2005) Automatic object trajectory based motion recognition using Gaussian mixture models. In: 2005 IEEE international conference on multimedia and expo (ICME). Amsterdam, Netherlands, pp 1532–1535
26. Bashir F, Qu W, Khokhar A, Schonfeld D (2005) HMM-based motion recognition system using segmented PCA. In: IEEE international conference on image processing (ICIP). Genova, Italia, pp 1288–1291
27. Johnson N, Hogg D (1996) Learning the distribution of object trajectories for event recognition. Image Vis Comput 14(8):609–615
28. Sumpter N, Bulpitt AJ (1998) Learning spatio-temporal patterns for predicting object behavior. Image Vis Comput 18(9):697–704
29. Lee JG, Han J, Whang KY (2007) Trajectory clustering: a partition-and-group framework. In: 2007 ACM SIGMOD international conference on management of data. Beijing, China, pp 593–604
30. Piciarelli C, Foresti GL (2006) On-line trajectory clustering for anomalous events detection. Pattern Recogn Lett 27(15):1835–1842
31. Wang X, Ma KT, Ng GW et al (2011) Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. Int J Comput Vision 95(3):287–312
32. Li J, Gong S, Xiang T (2008) Global behaviour inference using probabilistic latent semantic analysis. In: British machine vision conference (BMVC). Leeds, UK, pp 193–202
33. J. Li, S. Gong, and T. Xiang (2008) Scene segmentation for behaviour correlation. In: 10th European conference on computer vision (ECCV). Marseille, France, pp 383–395
34. Li J, Gong S, Xiang T (2012) Learning behavioural context. Int J Comput Vision 97(3):276–304
35. Zhou B, Wang X, Tang X (2011) Random field topic model for semantic region analysis in crowded scenes from tracklets. IEEE computer society conference on computer vision and pattern recognition (CVRP). CO, USA, pp 3441–3448

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.