An unsupervised approach for traffic motion patterns extraction

Amin Moradi | Asadollah Shahbahrami | Alireza Akoushideh

1 Department of Computer Engineering, Faculty of Engineering, University of Guilan, Rasht, Iran
2 Shad-Chamran College, Guilan Branch, Technical and Vocational University, Rasht, Iran

Correspondence
Asadollah Shahbahrami, Department of Computer Engineering, Faculty of Engineering, University of Guilan, Rasht, Iran.
Email: shahbahrami@guilan.ac.ir

Abstract

Automatic analysis, understanding typical activities, and identifying vehicle behaviour in crowded traffic scenes are fundamental and challenging tasks for traffic video surveillance. Some recent researches have been using machine learning approaches to extract meaningful patterns occurring in a traffic scene, for example, intersection. In this regard, we convert visual patterns and features to visual words using dense and sparse optical flow and learning traffic motion patterns with group sparse topical coding (GSTC) algorithm. In the first step of the proposed algorithm, the input traffic video is divided into non-overlapping clips. After that, motion vectors are extracted using dual TV-L1 as a dense optical flow and Lucas–Kanade as a sparse optical flow and converted to flow words. For learning traffic motion patterns, the GSTC algorithm, that is, a non-probabilistic topic model (TM) has been applied. These patterns represent priors on observable motion, which can be utilised to describe a scene and answer behaviour questions such as what are the motion patterns in a traffic scene and what is going on. The experimental results which have been obtained using a real dataset, QUML, show that the combination of the GSTC + dual TV-L1 extracts more traffic motion patterns in comparison with the GSTC + Lucas–Kanade and previous studies.

1 | INTRODUCTION

Surveillance cameras are a common way to increase security and improve management in public places because they can provide all information and events of the scene to decision makers. These cameras are used in almost all public places, and intelligent transportation systems (ITS) [1]. In ITS, these cameras can be used to monitor and control the traffic count, type of traffic, vehicle violations, and traffic flow at intersections and roundabout. On the one hand, the use of these video surveillance cameras has become widespread. On the other hand, with the expansion of these cameras, large amounts of multimedia data have been created, which effectively makes it impossible for managers to monitor and to control. So, it requires an automated analysis system to understand the typical activities and to identify the traffic motion patterns [2].

In urban traffic, the intersections are critical points. Any occurrence in the intersections when not made out of regular traffic can affect the total traffic of the city. Understanding the ongoing events and answering the behaviour questions such as ‘what is going on in this scene?’ is a fundamental task for machine vision systems. Extracting the general traffic motion patterns and features using the surveillance cameras are the first step to smooth the way to the state-of-the-art machine vision system for controlling traffic. General traffic motion patterns show the common activities of a scene that can be used for other important tasks such as rule mining and violation detection.

Discovering the traffic motion patterns result in a meaningful model of the scene and facilitate scene analysis. To extract the traffic motion patterns and behaviour understanding, researchers have developed new analysis approaches based on machine learning [3]. In traffic scenes such as intersection and roundabout, some of the usual activities, called motion scene models, happen regularly and periodically. By observing motion over time, the typical motion patterns can be learned and used as a priori knowledge for prediction. The advantage of these techniques is that they are generally applicable and do not require manual retraining for new scenes or scenarios. There are two particular methods for the automatic traffic behaviour understanding in this field: (1) Trajectory clustering and (2) topic modelling. In the first approach, moving objects such as vehicles are observed over a long period. Trajectories are
extracted from the video during observation to build a training database. The training trajectories are then clustered into groups of similar trajectories. These clusters represent the prototypical patterns of motion encountered in a scene. However, the quality of those methods is highly dependent on the robust tracking of vehicles, which is inherently difficult in general due to noise, changing lighting conditions, shadows, and occlusion. In the second method, low-level features such as moving or general pixel movements are computed with optical flow algorithms, combined with topic models (TMs) and sparse coding [4, 5] for reaching the general perspective of the scene [3, 6–8]. This study focuses on the second approach for automatic traffic behaviour understanding.

For automatic traffic behaviour understanding purpose, the video taken from the surveillance cameras is first divided into a sequence of non-overlapping clips with the same period. Motion vectors in each clip are extracted using an optical flow algorithm and converted into visual words called flow words. To extract latent patterns that represent the common motion distribution in the scene, the group sparse topical coding (GSTC) [9] framework as one of the non-probabilistic TMs is applied. To compare the performance of optical flow algorithms, we use two different optical flow algorithms to extract motion vectors, Lucas–Kanade [10] as a sparse optical flow and dual TV-L1 [11] as a dense optical flow. Our main attempt is to test the combined model of the TM and the optical flow in extracting the common traffic motion patterns at the intersection. In other words, we are looking for an answer to the question “How many of the traffic motion patterns that are occurring at the intersection can be extracted in an unsupervised approach?”. We have combined two different models of the optical flow algorithms with the original GSTC and compared the extracted patterns from these two methods with the visible visual patterns in the dataset. Our experimental results show that the GSTC + dual-TV-L1 extracts more traffic motion patterns compared to the GSTC + Lucas–Kanade. In other words, the combination of the GSTC and dense optical flow extract more traffic motion patterns than sparse optical flow and compared to other related work. This performance is not free because the dense optical flow is computationally intensive than the sparse optical flow. In the dense optical flow, all pixels in the frames are processed, while in the sparse optical flow, a selected number of pixels are processed. Our study is inspired by [1, 3, 12] with the main difference as follows. In our study, traffic motion patterns of vehicles are extracted using original group sparse topical coding with two different dense and sparse optical flow algorithms. In improved GSTC, a new regularisation term has been added to the GSTC for sparse word representations. We have not applied sparsity in our TM.

The remainder of the study is organised as follows: Section 2 provides an overview of other methods on traffic and behaviour analysing and background about the TM and optical flow. Section 3 gives a brief description of the problem. Section 4 describes traffic motion patterns detection using the TM and optical flow. In Section 5, the implementation and results are described. Finally, the conclusion is outlined in Section 6.

### Table 1: The abbreviation which have been used

| Acronym   | Full representation                          |
|-----------|----------------------------------------------|
| GSTC      | Group sparse topical coding                  |
| L-K       | Lucas–Kanade sparse optical flow algorithm   |
| Dual TV-L1| Dual total variation ℓ1 norm dense optical flow algorithm |
| ITS       | Intelligent transportation systems           |
| TM        | Topic model                                  |
| NLP       | Natural language processing                  |
| PLSA      | Probabilistic latent semantic analysis       |
| LDA       | Latent Dirichlet allocation                  |
| HDP       | Hierarchical Dirichlet process               |
| STC       | Sparse topical coding                        |

## 2 | BACKGROUND AND RELATED STUDY

### 2.1 | Background

#### 2.1.1 | TM

In order to make it easy to follow the content of the study, we have provided all used abbreviations in Table 1. TM was first developed in the context of text mining and natural language processing (NLP) to extract the latent topic that occurs in a collection of documents. TMs such as probabilistic latent semantic analysis (PLSA) [13], latent dirichlet allocation (LDA) [14], hierarchical Dirichlet process (HDP) [15], and sparse topical coding (STC) [16] recognise actions by exploring the co-occurrence of features at different hierarchical levels. The GSTC has been proposed as a new non-probabilistic TM to learn the sparse latent representations of large collections of text documents [9].

Suppose a collection of documents $D$ in the number of $M$ is given, containing words from a vocabulary $V$ in the number of $N$, a whole collection of documents is $D \in \mathbb{Z}^{N \times M}$, in which $d_{n,m}$ denotes the number of appearances of the $n$th word in the $m$th document. Let $\beta \in \mathbb{R}^{K \times N}$ be a dictionary with $K$ bases, where each base is assumed to be a topic base, that is, a unigram distribution over $V$. For documents $D$, GSTC projects $D$ into a semantic space spanned by a set of automatically learned topic bases and directly obtain the unnormalised word code $\beta \in \mathbb{R}^{K \times N \times h}$ for each individual word in documents $D$ [9]. Table 2 represents the notations used in the GSTC algorithm.

The GSTC solves the optimisation problem according to Equation (1). The first part of Equation (1) is equivalent to minimising an unnormalised KL-divergence between observed word counts $d_{n,m}$ and their reconstructions $S_{n,m,k} \beta_{k,r}$. Variable $S_{n,m,k}$ denotes the $n$th row of word code of $m$ document (clip) and $\beta_{k,r}$ denotes the $r$th column of $\beta$. The second part is a group lasso, that is, a mixed $\ell 1/\ell 2$ norm [9]. Equation (2)
TABLE 2 Different notations which has been used in the GSTC algorithm

| Notation | Description |
|----------|-------------|
| $D$ | Collection of documents |
| $M$ | Number of documents |
| $K$ | Number of topics |
| $V$ | Vocabulary set |
| $N$ | Size of vocabulary |
| $s$ | The dictionary of topics |
| $S$ | The word code representation of documents |
| $\lambda$ | Non-negative hyperparameter |

shows the linear algebra operation form of Equation (1):

$$
\begin{align*}
\min \mathcal{L}(S, \beta) &= \min_{\beta} \sum_{m=1}^{M} \sum_{n=1}^{N} \left( \sum_{k=1}^{K} S_{m,n,k} \beta_{k,n} - d_{m,n} \ln \left( \sum_{k=1}^{K} S_{m,n,k} \beta_{k,n} \right) \right) \\
&+ \lambda \sum_{m=1}^{M} \sum_{k=1}^{K} \left| \beta_{k,n} \right|_2 + C \\
\text{s.t. } S_{m,n,k} &\geq 0 \quad \forall m, n \\
\sum_{n=1}^{N} \beta_{k,n} &= 1, \quad \forall k \\
\end{align*}
$$

(1)

where $S_{m,n,k} \in \mathbb{R}^{N \times K}$ is a word code of $m$th document, $D_{m,n}$ is a $m$th document and $||\cdot||_{2,1}$ is a mixed $\ell 1/\ell 2$ norm.

2.1.2 Optical flow

A motion vector is a 2D vector that provides an offset from the coordinates in the current frame to the coordinates in a reference frame. The motion vector shows pixel or object displacement in consecutive frames of a video. The optical flow is one of the algorithms to compute the motion vector. In other words, optical flow is the motion of pixels or objects between consecutive frames in a video, caused by the relative movement between the object and camera. Optical flow algorithms are divided into two groups, dense and sparse. The goal of both algorithms is the same, that is, computing pixels displacement, but basically they are different in the mathematical formulation and hypotheses. In general, the algorithm that calculates consecutive frame displacements for a subset of pixels is called sparse optical flow, and the algorithm that calculates consecutive frame displacements for all pixels is called dense optical flow. The sparse optical flow algorithm has three main inputs: The current frame, a subset of pixels in the current frame, and the next frame. While the dense optical flow algorithm has two main inputs: The current and the next frames. [10, 11, 17, 18].

2.2 Related study

Most existing methods for analysing traffic can be divided into two categories. In the first category called ‘trajectory clustering’, the object such as a vehicle or pedestrian is detected and observed over a long time. Trajectories are extracted from the video using low-level motion or appearance features such as optical flow. In the second category, the motion of video frames is extracted without exact detection and tracking algorithms. Then, directly from these extracted properties, a model of motions and activities is constructed [3]. In [21], a statistical model of optical flow features was employed to obtain a representation for the motion patterns. In [22, 23], typical activities were characterised by hierarchical Bayesian models such as LDA and HDP. Dynamic dual-HDP non-parametric Bayesian model was used in [24] to automatically model activity categories and semantic regions without specifying the number of topics and with the online updates of the model. In [25], the two-level LDA TM learned first for single-agent motion, which is input to second level LDA for multi-agent interactions. Fast rank-1 robust PCA is used for foreground detection with counts of pixels in blocks used as input for DPMM learning, enabling incremental learning and inference [26]. In [1, 12, 27], LDA and sparse coding were used for efficient learning, scene understanding, and abnormal event detection. A two-staged cascaded LDA model was formulated in [28] for automatic discovering and learning of behavioural context for video-based complex behaviour recognition and anomaly detection. The study in [29] relied on PLSA to identify abnormal activities and repetitive cycles while [30] used an LDA to detect usual and unusual activities in the traffic scene. A topic-related STC is proposed for abnormality detection in the traffic patterns [27]. In [31], semi-supervised method based on sparse TM was proposed to detect anomalies in video surveillance. Table 3 shows some important work in traffic analysis with TM.
### TABLE 3  A comparison of some recent related work in traffic analysing

| Algorithm   | Input data                                                                 | Dataset                                                                 | Description                                                                                                                                                                                                 | Result                                      |
|-------------|----------------------------------------------------------------------------|-------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|
| PLSA [29]   | Foreground pixels with Lucas–Kanade optical flow                           | 45 min of video captured from a busy traffic                             | Unsupervised learning approach relying on PLSA applied to a rich set visual features including motion and size activities for discovering relevant activity patterns occurring in such scenes                                      | Abnormality detection                       |
| LDA [30]    | Foreground pixels with optical flow                                         | 1. QMUL Intersection dataset                                             | The dynamic TM, termed a Markov clustering TM. This model profiles complex dynamic scenes by robustly clustering visual events into activities and these activities into global behaviours with temporal dynamics                              | Unsupervised modelling scene behavioural characteristics underpinned at different spatial and temporal levels; online behaviour screening and saliency detection; online behaviour classification and semi-supervised detection of specified interesting behaviours |
|             |                                                                           | 2. Pedestrian crossing dataset                                            |                                                                Grid pdf                                                                find a set of visual features including motion and size activities heavy with activity patterns occurring in such scenes                                                                    |                                             |
|             |                                                                           | 3. Subway platform dataset                                                |                                                                Grid pdf                                                                find a set of visual features including motion and size activities heavy with activity patterns occurring in such scenes                                                                    |                                             |
| Two-level LDA | Moving pixels with optical flow [25]                                      | MIT traffic dataset                                                      | A two-level motion pattern mining approach is proposed. At the first level, single-agent motion patterns are modelled as distributions over pixel-based features. At the second level, interaction patterns are modelled as distributions over single-agent motion patterns                          | Discover both single-agent motion patterns and interaction patterns in the video.                                                                                   |
| HDP [23]    | Dual TV-L1 optical flow                                                   | 1. Public scenes in Zurich                                               | The first method extracts rules based on a learned TM, the second model is a Markov clustering TM. This model profiles complex dynamic scenes by robustly clustering visual events into activities and these activities into global behaviours with temporal dynamics | Spatio-temporal dependencies of moving agents in dynamic scenes                                                                                                           |
|             |                                                                           | 2. QMUL datasets                                                         |                                                                                                                                         |                                             |
| DPMM [26]   | Foreground pixels                                                          | 1. Freeway intersection video                                             | Fast rank-1 constrained robust PCA to compute the foreground, with non-parametric Bayesian models for inference                                                                                      | Anormal activity detection and patterns were discovered                                                                                                               |
|             |                                                                           | 2. Train station video                                                    |                                                                                                                                         |                                             |
|             |                                                                           | 3. Street surveillance video                                              |                                                                                                                                         |                                             |
| STC [1]     | Dual TV-L1 optical flow                                                   | QMUL intersection dataset                                                | An unsupervised approach to automatically explore motion patterns occurring in dynamic scenes under an improved STC framework                                                                        | Traffic pattern detection and abnormality detection.                                                                                                                     |
| GSTC [12]   | Lucas–Kanade optical flow                                                 | QMUL intersection Dataset                                                | An improved GSTC For scene analysing                                                                                                             | Abnormality detection                       |
| Topic-related STC [27] |                           | QMUL intersection dataset                                                | A topic-related STC is proposed in this study for analysing video sequences of traffic surveillance                                          | Traffic patterns, phases and abnormality detection                                                                                                                     |
| Semi-supervised sparse TM [34] | Trajectory extracted with Fisher kernel                                   | QMUL and AVSS dataset                                                    | Semi-supervised method based on sparse TM is proposed to detect anomalies in video surveillance                                                                          | Abnormality detection                       |

3  | TRAFFIC MOTION PATTERNS

In intersections, some of the usual activities called traffic motion patterns happen regularly and periodically. Actions such as turning left, turning right, and passing through intersection occur almost in all intersections. Twelve usual traffic motion patterns and their descriptions are depicted in Figure 1 and Table 4.

4  | AUTOMATIC TRAFFIC MOTION PATTERNS DETECTION

In the traffic video, there are visual features such as vehicle motion and motion patterns in different directions. For extracting traffic motion patterns, the occurrence of visual features and activities in clips should be converted to a predefined
bag-of-words. Then the extracted words are categorised based on their features, and final motion patterns are detected. The general block diagram of the proposed approach for traffic motion pattern detection is depicted in Figure 2. There are three main steps. In the first step, the input traffic video is divided into a set of non-overlapping subclips. Then motion vectors of each clip are calculated with dense and sparse optical flow for each consecutive frame in clips and converted into flow words in the second step. Finally, histograms of flow words represented in the second step are converted to a matrix called bag-of-words and given to the GSTC model for extraction of traffic motion patterns. In the following Section 4.1, each step is explained in detail.

### 4.1 Converting to short clips

In the proposed approach, the TM has been used. The TM treats a web page or single article as a document in NLP and text mining, while our data is video. Therefore, the input video should be considered as a collection of documents. There are different words in each document, while in video frames and clips, there are different visual features. Based on this, there is a mapping strategy from video to document representation. The whole video is divided into non-overlapping subclips and are
4.2 Modelling traffic flows

Pixels displacement is a key feature to detect general activities between video frames. Hence, this property is considered as a key feature to detect traffic motion patterns automatically. Pixels displacement is raw data and it should be converted into a fixed vector to use in the learning algorithm. The left-hand side of Figure 3 depicts the eight, north, northeast, northwest, south, southeast, southwest, east, and west directions for each pixel displacement for two consecutive frames. These different directions are converted into a fixed vector as depicted in Figure 3. All pixel displacement which is located into cells or blocks is converted into a fixed vector \((R^N)\) of size \(N = \text{number of cells} \times 8\). Each entry in vector shows the number of occurrences of specific directions in that block. In the following subsections, modelling of traffic flows with the dense and sparse optical flow are described.

4.2.1 Modelling traffic flows with dense optical flow

The dense optical flow provides a motion vector per pixel in each of the two consecutive frames. Figure 4 depicts the flowchart of flow word extraction with the dual TV-L1 algorithm. We first give each two consecutive frames to the dual TV-L1 algorithm.

The motion vectors matrix is divided into \(C_x \times C_y\) square cells, each of which contains \(p \times p\) pixels. A threshold is applied to motion vectors to eliminate noise and maintain reliable flows.

To generate the flow words, we sample the remaining motion vectors, \(s_j = (x, y, u, v)\), where \((x, y)\) is positioned on a grid with \(p\) pixels spacing. Then the variables \((u, v)\) of the motion vector of the samples are quantised into eight directions.

4.2.2 Modelling traffic flows with sparse optical flow

The sparse optical flow provides a motion vector for a set of predetermined pixels that has interesting features such as edges, corners, and textures. The flowchart of flow word extraction with the Lucas–Kanade algorithm is illustrated in Figure 5. For every two consecutive frames, at first, we find key points using Shi and Tomasi corner detector [32]. Then, frames are divided into \(C_x \times C_y\) square cells, each of which contains \(p \times p\) pixels. The vector motion of the key point is computed with the Lucas–Kanade algorithm. Again, a threshold is applied to motion vectors to eliminate noise and maintain reliable flows. Motion vectors are denoted by \((x, y, \alpha)\). The positions \((x, y)\) are quantised to the nearest position on a grid of \(p\) pixel spacing and the angles of motion vectors, \(\alpha\), are quantised into eight directions.

Finally, a set of fixed vocabulary \(V = \{1, \ldots, N\}\), \(N = C_x \times C_y \times 8\) is formed. Each word contains two aspects of the content: Location and direction information. Flow words of video clips are collected all over the frames. Then a video clip is represented as a vector \(d_w = (1, \ldots, N)^T\), which each entry in \(d_w\) specifies the number of occurrences of flow word \(n\) in this clip. In this study, we define a motion pattern as a spatial distribution of flow words. Motion patterns, denoted as \(\beta\), are corresponding to the latent topics in a TM. Each row of \(\beta\) is a topic basis, which is a distribution over the vocabulary \(V\), i.e., \(\beta_k \in P\), where \(P\) is a \((N - 1)\)-simplex.

4.3 Learning traffic motion patterns

The learning traffic motion pattern step aims to cluster a co-occurrence flow word. At first, all the flow words extracted from the previous step transform into a matrix called bag-of-words. Each column is a single clip and its flow words. This matrix is input to the learning algorithm. As can be seen in the last part of Figure 2, learning traffic motion patterns, flow word codes are extracted from the bag-of-words matrix, then the extracted flow
word codes are clustered. In the flow word coding step, each column of the bag-of-words matrix transforms into a matrix that shows the portion of observed flow word in each motion pattern. In the learning patterns step, all the computed flow word codes cluster to a $K$ motion pattern that shows general traffic motion patterns of the scene.

The GSTC uses group lasso and selects connected and correlated variables instead of individual variables. It achieves this by applying a mixed $\ell_1/\ell_2$ norm (the sum ($\ell_1$ norm) of the $\ell_2$ norms) over different groups of variables. In our study, each traffic motion pattern is a group of flow words, so this feature of the GSTC is suitable for choosing a group of flow words as a traffic motion pattern. This feature and traditional sparse representation are very suitable for combining the TM and optical flow, especially in the case of GSTC + dual TV-L1, because dense optical flow computes all pixels displacement and we need a sparse representation of flow word and choosing group of flow words as a traffic motion pattern. In the case of GSTC...
Lucas–Kanade, just the correlated variables selected jointly feature was worked.

### 4.3.1 Learning algorithm formulation

A typical solution to solve objective function in Equation (2) is coordinate descent algorithm (CDA) [33] because it is a bi-convex function, that is, convex over either \( \{s_m\}_{m=1}^M \) or \( \beta \) when the other is fixed. The CDA alternatively performs the optimisation over \( \{s_m\}_{m=1}^M \) and \( \beta \) as depicted in Figure 6:

\[
\min \mathcal{L}(s, \beta) = \min_{s, \beta} \sum_{m=1}^M \left( \left\| D_{m,\ast} - \text{diag}(s_m) \right\|_2^2 + \lambda \| s_m \|_{2,1} \right)
\]

subject to \( s_m \geq 0 \quad \forall m, \quad \beta_k \geq 0, \quad \| \beta_k \|_1 = 1 \quad \forall k \) (3)

In Equation (3), like the GSTC formulation in [12], the \( \ell_2 \) norm of the reconstructions error is minimised.
4.3.2 Learning flow word code

This step aims to find the portion of the extracted flow words of the clip in each motion pattern. Cube $S$ in Figure 2 depicts the flow word code of the clips. As mentioned, the objective function is bi-convex. So, the algorithm optimises the $S$ when dictionary $\beta$ is fixed. Since the word codes are grouped in topics and each group is separable, the block coordinate descent (BCD) method is baptised for this optimisation in [9, 12] by solving the problem for each $S_{nk}$ alternatively as formula in Equation (4):

$$
\min \left\llbracket \| D_{n,m} - \text{diag}(S_{n,k}) \|_2^2 + \lambda \| S_n \|_{2,1} \right\rrbracket \quad \text{s.t.} \quad s > 0
$$

(4)

Instead of calculating each $S_{nk}$ separately, Equation (6) is transformed to Equation (7) for solving entire matrix $\beta$:

$$
\min \sum_{m=1}^{M} \left( \| D_{n,m} - \text{diag}(S_{n,k}) \|_2^2 \right) = \sum_{m=1}^{M} \sum_{n=1}^{N} \left( D_{n,m} - S_{n,k} \beta_{k,n} \right)^2
$$

s.t. $\beta \geq 0$, $\sum_{n=1}^{N} \beta_{k,n} = 1$, $\forall k$ \hspace{1cm} (6)

In Equation (5), $T$ is a diagonal matrix of document $D_{n,m}$ and $R = \text{diag}([1/\| S_{1} \|_2^2 + \varepsilon, \ldots, 1/\| S_{K} \|_2^2 + \varepsilon])$. The encoding algorithm is depicted in Figure 8.

4.3.3 Learning motion patterns

In this step, all the extracted flow word codes of the clips cluster to a similar $K$ group in matrix $\beta$. Each row of $\beta$ is a unigram distribution of flow words that shows general traffic motion patterns of the scene. The matrix $\hat{\beta}$ is updated by solving the optimisation problem in Equation (6):

$$
\min_{\beta} \sum_{m=1}^{M} \left( \| D_{n,m} - \text{diag}(S_{n,k}) \|_2^2 \right) = \sum_{m=1}^{M} \sum_{n=1}^{N} \left( D_{n,m} - S_{n,k} \beta_{k,n} \right)^2
$$

s.t. $\beta \geq 0$, $\sum_{n=1}^{N} \beta_{k,n} = 1$, $\forall k$ \hspace{1cm} (6)

Instead of computing each $\beta_{k,n}$ separately, Equation (6) is transformed to Equation (7) for solving entire matrix $\beta$:

$$
\sum_{d=1}^{D} \text{trace} \left( TT - S_{n} \beta^{T} S_{n}^{T} \right) = \text{trace} \left( S_{n} \beta^{T} S_{n}^{T} \right)
$$

(7)

where $\beta^{T}$ and $S_{n}^{T}$ is transpose matrix and $T$ is a diagonal matrix of document $D_{n,m}$. The encoding algorithm is shown in Figure 7.

5 IMPLEMENTATION RESULTS

5.1 Environmental setup and dataset

The proposed approach was implemented in the visual studio C++ programming environment. The OpenCV [34] functions were used for computing optical flow for Lucas–Kanade and dual TV-L1 algorithms. Linear algebra C++ library (armadillo
Hardware system was a PC with Intel Core i7 4790 CPU and 16GB RAM. Performance evaluation of the proposed approach was performed using the QMUL junction video dataset [36]. Figure 9 depicts one sample frame of this dataset, which has been captured at 25 frames per second with 360 × 288 resolution in an intersection in London. In this dataset which contains 90,000 frames, the density of vehicles is relatively high and therefore motions are quite complex. Moreover, this video dataset has quickly become a favourite dataset for automatic traffic analysis. Twelve traffic motion patterns and their description have already been mentioned in Table 4 and Figure 1. Nine of these patterns exist in the QMUL dataset which is depicted in Table 5. It should be noted that the angle of the camera and its height from the ground has an important role in extracting the correct and exact motion vector. The 200 and 100 clips have been used in GSTC + dual TV-L1 and GSTC + Lucas–Kanade methods, each containing 75 and 150 frames, respectively.

### Table 5: Visibility motion patterns in QMUL dataset (according to Figure 1)

| Motion pattern | Motion pattern name | Visibility |
|----------------|---------------------|------------|
| 1              | Turning left from the south edge to the west |✗ |
| 2              | Passing through an intersection from the east edge to the west | ✓ |
| 3              | Turning right from the north edge to the west | ✓ |
| 4              | Turning left from the west edge to the north | ✓ |
| 5              | Passing through an intersection from the south edge to the north | ✓ |
| 6              | Turning right from the east edge to the north | ✓ |
| 7              | Turning left from the north edge to the east |✗ |
| 8              | Passing through an intersection from west edge to the east | ✓ |
| 9              | Turning right from the south edge to the east | ✓ |
| 10             | Passing through an intersection from north edge to the south | ✓ |
| 11             | Turning right from the west edge to the south | ✓ |
| 12             | Turning left from the east edge to the south |✗ |

#### 5.2 Parameter settings

The whole video has been divided into 3-s length for GSTC + dual TV-L1 and 6-s length for GSTC + Lucas–Kanade sub-clips, based on that there are a different number of clips for both algorithms. For both optical flow algorithms, the scene has been divided into 10 × 10 cells. Obtained motion vectors are quantised to one of eight directions (45°, 90°, 135°, 180°, 225°, 270°, 315°, 360°). Therefore, according to this video dataset, the size of set $V$ is equal to $N = 36 \times 29 \times 8$. The hyper-parameter $\lambda$ experimentally set to 1.0 in case of using dual TV-L1 and 0.5 in case of using Lucas–Kanade. The maximum iteration of the learning algorithm was considered 20.
and the parameter $\varepsilon$ sets to 0.01 for checking convergence. Without any prior knowledge, usually, a large value is considered for $K$. We set the number of clustering, $K = 20$ in our experiments.

### 5.3 Experimental results

Table 6 depicts the obtained results of both GSTC + dual TV-L1 and GSTC + Lucas–Kanade methods. Both algorithms extract nine exiting visual features in the dataset, while GSTC + dual TV-L1, which uses a dense algorithm, extracts more visual patterns than the other algorithm. In the intersection, in each frame and clip, there may be a combination of different visual features. This is because vehicles pass in different directions simultaneously from a fixed camera position point of view. In the dense algorithm, all pixels are considered in the calculation; more combinations of traffic patterns are considered while this behaviour is not seen in the sparse algorithm. There are different, one, two, and three motion patterns in 17 clusters from 20 clusters in GSTC + dual TV-L1, while there are 12 clusters in the second algorithm. In other words, the first algorithm extracts more traffic motion patterns than the second algorithm which uses sparse optical flow.

| $K$ | GSTC + dual TV-L1(# of motion pattern in Table 4) | GSTC + Lucas–Kanade(# of motion pattern in Table 4) |
|-----|--------------------------------------------------|--------------------------------------------------|
| 1   | –                                                | 2, 6                                             |
| 2   | 6                                                | –                                                |
| 3   | 9                                                | 4, 8, 11                                         |
| 4   | 2                                                | 5, 10                                            |
| 5   | –                                                | –                                                |
| 6   | –                                                | –                                                |
| 7   | 5, 10                                            | –                                                |
| 8   | 4, 11                                            | 8                                                |
| 9   | 8, 11                                            | 10                                               |
| 10  | 5, 9                                             | 3, 9                                             |
| 11  | 2, 6                                             | 4, 11                                            |
| 12  | 5, 9                                             | 3, 10                                            |
| 13  | 8                                                | 6                                                |
| 14  | 11                                               | 4, 8                                             |
| 15  | 4, 8                                             | 5, 9                                             |
| 16  | 4, 8, 11                                         | 5                                                |
| 17  | 10                                               | –                                                |
| 18  | 5, 10                                            | –                                                |
| 19  | 6, 2                                             | –                                                |
| 20  | 3, 4                                             | –                                                |

Some extracted traffic motion patterns using GSTC + dual TV-L1 and GSTC + Lucas–Kanade are depicted in Figures 10 and 11, respectively. As mentioned, dense optical flow provides motion vectors for all pixels which causes more flow words obtained from clips than sparse optical flow. Therefore, more meaningful motion patterns were detected.

A comparison of our results using the GSTC + dual TV-L1 and GSTC + Lucas–Kanade algorithms with improved STC with dual-TV-L1 approach [1] and improved GSTC with Lucas–Kanade optical flow [12] is presented in Table 7. Our approach extracts more meaningful traffic motion patterns. For example, motion pattern number 11 in Figure 10(d) has been detected while not detected in Figure 12 using STC and dual TV-L1 [1]. As another example, more motion patterns such as #3 and #6 in Figures 11(a) and (b) have been extracted compared to just four motion patterns detected in Figure 13 using improved GSTC and Lucas–Kanade [12]. Besides, we are trying to answer this question in an intersection: ‘What is going on?’ using extracting meaningful traffic motion patterns. While in [1, 12], motion patterns have been used for abnormality detection, a clip was classified as normal or abnormal. A rough comparison of our study and [1, 12] is depicted in Table 7. In fact, Table 7 is a comparison between visible motion patterns in the dataset (Table 5) and extracted motion patterns with both methods (Figures 10 and 11) with respect to defined common patterns (Figure 1 and Table 4). In other words, Figure 1 shows the possible patterns in the dataset, in which Table 5 indicates the number of visible motion patterns. Studies in [1, 12] have used these motion patterns for abnormality detection. They classified a clip as normal or abnormal, and if a clip cannot be described with obtained patterns, it is considered as an anomaly.

| Motion pattern | Improved STC with dual TV-L1 [1] | Improved GSTC with Lucas–Kanade [12] | GSTC with Lucas–Kanade | GSTC with dual TV-L1 |
|----------------|----------------------------------|---------------------------------------|------------------------|----------------------|
| 2              | ✓                                | ✓                                     | ✓                      | ✓                    |
| 3              | ✓                                | ✗                                     | ✓                      | ✓                    |
| 4              | ✓                                | ✓                                     | ✓                      | ✓                    |
| 5              | ✓                                | ✓                                     | ✓                      | ✓                    |
| 6              | ✓                                | ✗                                     | ✓                      | ✓                    |
| 8              | ✓                                | ✓                                     | ✓                      | ✓                    |
| 9              | ✓                                | ✓                                     | ✓                      | ✓                    |
| 10             | ✓                                | ✗                                     | ✓                      | ✓                    |
| 11             | ✗                                | ✓                                     | ✓                      | ✓                    |
| 8              | 4                                | 9                                     | 9                      |                      |

Table 6 Performance evaluation in terms of the number of extracted motion patterns of GSTC + dual TV-L1 and GSTC + Lucas–Kanade which use dense and sparse optical flow algorithms, respectively.

Table 7 Comparison of our study with other related studies (comparing the performance of our unsupervised method with visible patterns and how many visible visual patterns has been extracted with the unsupervised method).
5.4 | Discussion

Although the combination of the TM and the dense optical flow yields more motion patterns due to the computing motion vector of each pixel, investigating and obtaining motion vectors of all available pixels impose a high computational cost. On the other hand, the combination of the TM and sparse optical flow has a low computational cost, while it extracts fewer motion patterns using the GSTC and Lucas–Kanade optical flow algorithm.
**FIGURE 12** Some extracted motion patterns using improved STC and dual TV-L1 [1]

**FIGURE 13** Some extracted motion patterns using improved GSTC and Lucas–Kanade [12]
patterns, and selecting pixels to calculate the motion vector is the challenging task. In other words, pixels displacement between frames as a feature and TM as a learning algorithm are used to extract traffic motion patterns, while it does not need object detection and tracking algorithms. To increase performance, the vehicle object should be detected in frames, for example, using the YOLO framework, the motion vector of detected objects is computed. It is possible to create a new bag-of-words matrix, where each column belongs to a single object in contrast to the common method that each column belongs to a clip. Combining object detection and sparse optical flow with clustering is our future study.

6 CONCLUSIONS

In urban traffic, intersections play an essential role. The traditional method of video surveillance is not suitable due to the huge amount of video data and the unreliability of the human being. It requires a system that automatically captures traffic motion patterns. In this study, an unsupervised approach was presented to extract traffic motion patterns and convert them into visual words in video surveillance. First, the whole traffic video dataset is divided into clips. The local motion vectors were extracted by dual TV-L1 as a dense optical flow for all pixels and Lucas–Kanade as a sparse optical flow for a set of specific pixels and then converted to flow words. Given the fact that at intersections, motions such as turning left, turning right, and passing through the intersection that are template patterns of traffic, we use the GSTC algorithm for learning traffic motion patterns. The experimental results show that the GSTC + dual TV-L1 extracts more traffic motion patterns than the GSTC + Lucas–Kanade algorithm and some previous studies.

DATA AVAILABILITY STATEMENT

Data derived from public domain resources: QMUL junction video dataset http://personal.ie.cuhk.edu.hk/~ccloy/downloads_qmul_junction.html http://www.eecs.qmul.ac.uk/~sgg/QMUL_Junction_Datasets/Junction/Junction.html

ORCID

Asadollah Shabbahrami DOI: https://orcid.org/0000-0002-5195-1688
Alireza Akoushideh DOI: https://orcid.org/0000-0001-9958-4613

REFERENCES

1. Fu, W., et al.: Dynamic scene understanding by improved sparse topical coding. Pattern Recog. 46(7), 1841–1850 (2013)
2. Fu, W., et al.: Learning semantic motion patterns for dynamic scenes by improved sparse topical coding. In: IEEE International Conference on Multimedia and Expo, pp. 296–301. IEEE, Piscataway (2012)
3. Morris, B.T., Trivedi, M.: Understanding vehicular traffic behavior from video: A survey of unsupervised approaches. J. Electron. Imaging 22(4), 041113 (2013)
4. Luo, F., et al.: Sparse-adaptive hypergraph discriminant analysis for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 17(6), 1082–1086 (2019)
5. Duan, Y., et al.: Local constraint-based sparse manifold hypergraph learning for dimensionality reduction of hyperspectral image. IEEE Trans. Geosci. Remote Sens. 1–16 (2020) https://doi.org/10.1109/TGRS.2020.295709
6. Datomdi, S.R.E., et al.: A survey of vision-based traffic monitoring of road intersections. IEEE Trans. Intell. Transp. Syst. 17(10), 2681–2698 (2016)
7. Shirazi, M.S., Morris, B.T.: Looking at intersections: A survey of intersection monitoring, behaviour and safety analysis of recent studies. IEEE Trans. Intell. Transp. Syst. 18(1), 4–24 (2016)
8. Wang, X.: Action recognition using topic models. In: Visual Analysis of Humans pp. 311–332. Springer, London (2011)
9. Bai, L., et al.: Group sparse topical coding: From code to topic. In: Proceedings of the 6th ACM International Conference on Web Search and Data Mining, pp. 315–324. ACM, New York (2013)
10. Patel, D., Upadhyay, S.: Optical flow measurement using Lucas–Kanade method. Int. J. Comput. Appl. 61(10), 6–10 (2013)
11. Zach, C., et al.: A duality based approach for realtime TV-L1 optical flow. In: Joint Pattern Recognition Symposium, pp. 214–223, Springer, Berlin (2007)
12. Ahmadi, P., et al.: Abnormal event detection and localisation in traffic videos based on group sparse topical coding. IET Image Proc. 10(3), 235–246 (2016)
13. Steyvers, M., Griffiths, T.: Probabilistic topic models. Thomas K., Lander, Danielle S., McNamara, Simon Dennis, Walter Kintsch. In: Handbook of Latent Semantic Analysis vol. 427(7), pp. 424–440. Routledge, New York (2007)
14. Blei, D.M., et al.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)
15. Teh, Y.W., et al.: Hierarchical Dirichlet processes. J. Am. Stat. Assoc. 101, 1566–1581 (2012)
16. Zhu, J., Xing, E.P.: Sparse topical coding. https://arxiv.org/ftp/arxiv/papers/1202/1202.3778.pdf (2012)
17. Fortun, D., et al.: Optical flow modeling and computation: A survey. Comput. Vision Image Understanding 134, 1–21 (2015)
18. Kuehler, A., Bradski, G.: Learning OpenCV 3: Computer Vision in C++. with the OpenCV Library. O'Reilly Media Inc., Sebastopol, CA (2016)
19. Morris, B.T., Trivedi, M.M.: A survey of vision-based trajectory learning and analysis for surveillance. IEEE Trans. Circuits Syst. Video Technol. 18(8), 1114, (2008)
20. Yang, Y., et al.: Video scene understanding using multi-scale analysis. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1669–1676. IEEE, Piscataway (2009)
21. Saleem, I., et al.: Scene understanding by statistical modeling of motion patterns. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2069–2076. IEEE, Piscataway (2010)
22. Wang, X., et al.: Unsupervised activity perception by hierarchical bayesian models. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE, Piscataway (2007)
23. Kuettel, D., et al.: What's going on? Discovering spatio-temporal dependencies in dynamic scenes. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1951–1958. IEEE, Piscataway (2010)
24. Wang, X., et al.: Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. Int. J. Comput. Vision 95(3), 287–312 (2011)
25. Song, L., et al.: Understanding dynamic scenes by hierarchical motion pattern mining. In: IEEE International Conference on Multimedia and Expo, pp. 1–6. IEEE, Piscataway (2011)
26. Rana, S., et al.: Large-scale statistical modeling of motion patterns: A Bayesian nonparametric approach. In: Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, p. 7. ACM, New York (2012)
27. Ahmadi, P., et al.: Employing topical relations in semantic analysis of traffic videos. IEEE Intell. Syst. 34(1), 3–13 (2018)
28. Li, J., et al.: Learning behavioural context. Int. J. Comput. Vision 97(3), 276–304 (2012)
29. Varadarajan, J., Odobe, J.-M.: Topic models for scene analysis and abnormality detection. In: 2009 IEEE 12th International Conference on
30. Hospedales, T., et al.: Video behaviour mining using a dynamic topic model. Int. J. Comput. Vision 98(3), 303–323 (2012)
31. Wang, J., et al.: Abnormal event detection with semi-supervised sparse topic model. Neural Comput. Appl. 31(5), 1607–1617 (2019)
32. Shi, J.: Good features to track. In: 1994 Proceedings of IEEE Conference On Computer Vision And Pattern Recognition, pp. 593–600 IEEE, Piscataway (1994)
33. Wright, S.J.: Coordinate descent algorithms. Math. Program. 151(1), 3–34 (2015)
34. Open source computer vision. https://opencv.org/
35. Armadillo C++. http://arma.sourceforge.net/
36. QMUL junction dataset. http://personal.ie.cuhk.edu.hk/~ccloy/downloads_qmul_junction.html

How to cite this article: Moradi A, Shahbahrami A, Akoushideh A. An unsupervised approach for traffic motion patterns extraction. IET Image Process 2021;15:428–442. https://doi.org/10.1049/ipr2.12033