BEHAVIOUR ANALYSIS USING A MULTI-LEVEL MOTION PATTERN LEARNING FRAMEWORK

Mohamed Gomaa Mohamed, Ph.D candidate (corresponding author)

Department of civil, geological and mining engineering

Polytechnique Montréal, C.P. 6079, succ. Centre-Ville

Montréal (Québec) Canada H3C 3A7

Phone: +1 (514) 340-5121 ext. 4210

Email: mohamed.gomaa@polymtl.ca

Nicolas Saunier, ing., Ph.D., Associate professor

Department of civil, geological and mining engineering

Polytechnique Montréal, C.P. 6079, succ. Centre-Ville

Montréal (Québec) Canada H3C 3A7

Phone: +1 (514) 340-4711 ext. 4962

Email: nicolas.saunier@polymtl.ca

Word count

Text	6073
Tables (0 X 250)	0
Figures (5 X 250)	1250
References	500
Total	7823

Date of submission: **November 15th , 2014**

ABSTRACT

The increasing availability of video data, through existing traffic cameras or dedicated field data collection, paves the way for the collection of massive datasets about the microscopic behaviour of road users using computer vision techniques. Analysis of such datasets helps to understand the normal road user behaviour, and it can be used for realistic prediction of future motion and computing surrogate safety indicators. A multi-level motion pattern learning framework is developed that enables automated scene interpretation, anomalous behaviour detection and surrogate safety analysis. Firstly, Points of interest (POI) are learnt based on Gaussian Mixture Model and the Expectation Maximization algorithm and then used to form activity paths (APs). Secondly, motion patterns, represented by trajectory prototypes, are learnt from road users' trajectories in each AP using a two-stage trajectory clustering method based on spatial then temporal (speed) information. Finally, motion prediction relies on matching at each instant partial trajectories to the learnt prototypes to evaluate the potential for collision by computing indicators. An intersection case study demonstrates the framework ability in many ways: it helps to reduce the computation cost up to 90 %, clean trajectories dataset from tracking outliers, use actual trajectories as prototypes without any pre- and post-processing and predict future motion realistically to compute surrogate safety indicators.

Keywords: motion patterns, unsupervised learning, scene interpretation, driver behaviour, surrogate measures of safety.

INTRODUCTION

The increasing availability of video data, through existing traffic cameras or dedicated field data collection, paves the way for the collection and analysis of massive datasets about the microscopic behaviour of road users using computer vision techniques. These datasets can be used in several transportation applications, e.g. identification of activity patterns, safety diagnosis, calibration and validation of macroscopic and microscopic models, and behaviour analysis at various space and time scales. New algorithms are therefore needed to mine these massive datasets. However, one of the main challenges of large datasets is computational cost.

For traffic safety analysis, diagnosis relies traditionally on historical collision data which have several shortcomings that have been repeatedly covered in previous work, e.g. in [1], [2]. Therefore, proactive methods based on the observation of all interactions that do not require waiting for accidents to happen have been advocated as surrogate or complementary approaches to diagnose road safety. The key defining concept of surrogate safety analysis is the collision course, i.e. a situation in which two road users would collide if their movements remain unchanged (taken from the conflict definition in [3]). This requires specifying a method to predict road users' motions in order to evaluate if they are on a collision course, and allows measuring several surrogate safety indicators such as the time to collision (TTC). Most current analyses rely on the rarely specified or justified method of motion prediction at constant velocity, while several possible paths may in general lead road users to collide. Moreover, this assumption is inadequate in many cases because it does not consider the road environment (e.g. intersection layout) and driver behaviour. This shortcoming is particularly visible for turning vehicles. An appropriate solution is to learn the frequent road user motions, known as motion patterns (MP), on the studied site. These patterns can then be used for motion prediction in a probabilistic manner to compute several surrogate safety measures.

This paper builds upon and extends past work, namely [1] and [4]. The work in [1] focused on motion prediction methods using simple kinematic assumptions: in this paper, the motion prediction method based on MPs is favored as it is more realistic and requires fewer assumptions. The work in [4] developed new similarity measures for time series data which were applied to surrogate safety indicators and are used in this paper to refine the learning of MPs using temporal (speed) information. New generic algorithms are proposed in this paper for multi-level motion pattern learning that enable automated scene interpretation, driver behavior understanding and anomalous behaviour detection and surrogate safety analysis. Moreover, logical constrains are proposed to accelerate the processing time for learning MPs and predicting road user motion for surrogate safety analysis. Road user trajectories are extracted from video data recorded with a fixed camera using a custom feature-based video tracker available in the open source "Traffic Intelligence" project [5]. This paper presents the following contributions:

1. An algorithm to reconstruct smooth road user trajectories based on feature trajectories.
2. The detection and exploitation of points of interest (POI), entry and exit zones, to discover static occlusion zones, filter trajectories, connect divided trajectories, detect outliers, and speed up MP learning and motion prediction.
3. MPs are extracted in two stages using spatial (positions) and temporal (speed) information. The integration of speed profiles in MPs is especially useful for motion prediction in the case of turning vehicles.
4. The application of the learnt prototypes (spatial and temporal) to anomaly detection, motion prediction and surrogate safety measures.
5. An open source software implementation [5] of the proposed methods to encourage adoption and further development.

The remainder of this paper is organized as follows: first the review of related work, then the presentation of the proposed method, followed by the experimental results from a real world case study of an intersection in Montreal, Canada, and finally the discussion of the results and conclusion.

PREVIOUS WORK

A variety of techniques were developed for the analysis of scene activities and the learning of motion patterns. Surveys of activity analysis were performed in [6, 7]. The activity analysis can be summarized in two main tasks; discovering POIs and learning activities. The first task refers to regions where some activity occurs, whereas the second task is to learn the activity paths which represent how the objects move between the POIs, which is known as trajectory learning.

There is relatively little work in the literature addressing the learning of POIs [8, 9, 10, 11, 12, 13], In [10], authors used the trajectory endpoints to construct the entry and exit datasets which are modeled using a Gaussian Mixture Model (GMM) and the Expectation Maximization (EM) algorithm. Two types of noise may be mixed with signal (actual interest points): noise caused by tracking failures and semi-stationary motion noise generated by sources such as waving trees and water reflections: these are usually learnt as a separate GMM (noisy clusters) that can be distinguished using a density criterion. The same methodology was applied to a simulated intersection dataset in [12]: the authors used the detected entry and exit zones to filter the trajectories dataset into complete and incomplete datasets. Only the complete trajectories were used in the motion learning phase, which may represent an important loss of some typical motion information (e.g. left turning users stopped in the intersection waiting for a gap in the opposite through traffic). Recently, Nedrich and Davis [13] modeled the entry and exit zones using “weak” trajectories, i.e. from trackers that provide multiple and frequently fragmented tracks per target. They applied a modified mean-shift clustering algorithm to cluster weak trajectory into higher level entities. The entity trajectories are then used to detect entry and exit regions based on the standard mean-shift clustering algorithm. Although the authors assert that mean-shift clustering is able to localize cluster modes automatically without knowing the number of clusters, the algorithm depends on a bandwidth criterion that is as challenging to select as the number of clusters. They used the detected zones to identify static occlusion regions based on an analysis of the distance distributions between the regions. From the available literature, it is evident that learning interest point and their transportation applications need further investigations.

On the other hand, the literature on trajectory learning is very rich. The objective of trajectory learning is to cluster a dataset of observed trajectories into the main subsets of similar trajectories or motion patterns. Bennewitz et al [14] studied MPs of people in a scene and updated the behaviour of a robot accordingly using the EM algorithm for clustering. Hu et al [15] modeled road user activities with a fuzzy self-organizing neural network. One of the main applications of this research is future motion prediction, which is highly influenced by the clustering. Another work by [16] proposed a system for learning motion pattern hierarchically using spatial and temporal information with the fuzzy K-means algorithm, where the MP are represented by a chain of Gaussian distributions and applied for anomaly trajectory and behaviour prediction. Atev et al [17] modeled traffic patterns in an intersection using a modified Hausdorff distance and spectral clustering.

Saunier et. al. [18] proposed a custom algorithm to learn MPs. This algorithm trades the parameter of the number of clusters for a maximum distance or minimum similarity between instances of the same cluster: when a new instance is too different from the existing clusters, a new one is created for it. The authors also proposed to use the original trajectories as representatives, or prototypes, in particular to use them for motion prediction. The last original idea was to favour “long” prototypes, in this case with long time periods of observation. This solves partially the problem of dependency of the results to the algorithm initialization which is a well-known challenge of many clustering algorithms such as k-means (initialization of the cluster centroids). A recent work by Morris and Trivedi [12] focused on automated activity learning in an unsupervised fashion. The authors proposed a three-stage hierarchal learning framework to analyze object activities and predict future activities, as well as to detect abnormal events. Like [18], the authors used the longest common subsequence similarity (LCSS) as the trajectory similarity measure but spectral clustering for trajectory clustering.

Previous work on speed profile clustering is limited to individual speed profiles, separate from motion patterns. Parkhurst [19] examined the shape of speed profiles to understand the driver behaviour at urban and rural non-signalized intersections. A typical speed profile was studied in case of a vehicle coming to a complete stop or failing to complete their stop at a stop sign intersection. Laureshyn et al [20] classified the speed profiles, extracted from automated video data, of vehicles making left turn at a signalized intersection. The left turn manoeuvre mainly interacted with two different types of road users: oncoming traffic and pedestrians at the pedestrian crossing. The authors used three types of pattern recognition techniques: k-means (unsupervised), k-nearest neighbors (supervised), and manual classification based on dimension reduction (singular value decomposition). They concluded on the superiority of the three pattern recognition techniques over human classification. Nevertheless, the joint or hierarchical learning of MPs and speed profiles and their use for behavior prediction have not been yet investigated.

Very recent work on behaviour learning has adapted the methods used for topic modelling in text document analysis such as measures of co-occurrence and bag-of-word representations using positions and velocities [21, 22]. These methods cannot be used for the purpose of motion prediction since they ignore the temporal order of positions in a trajectory and cannot characterize the individual activities (but provide a more holistic scene description).

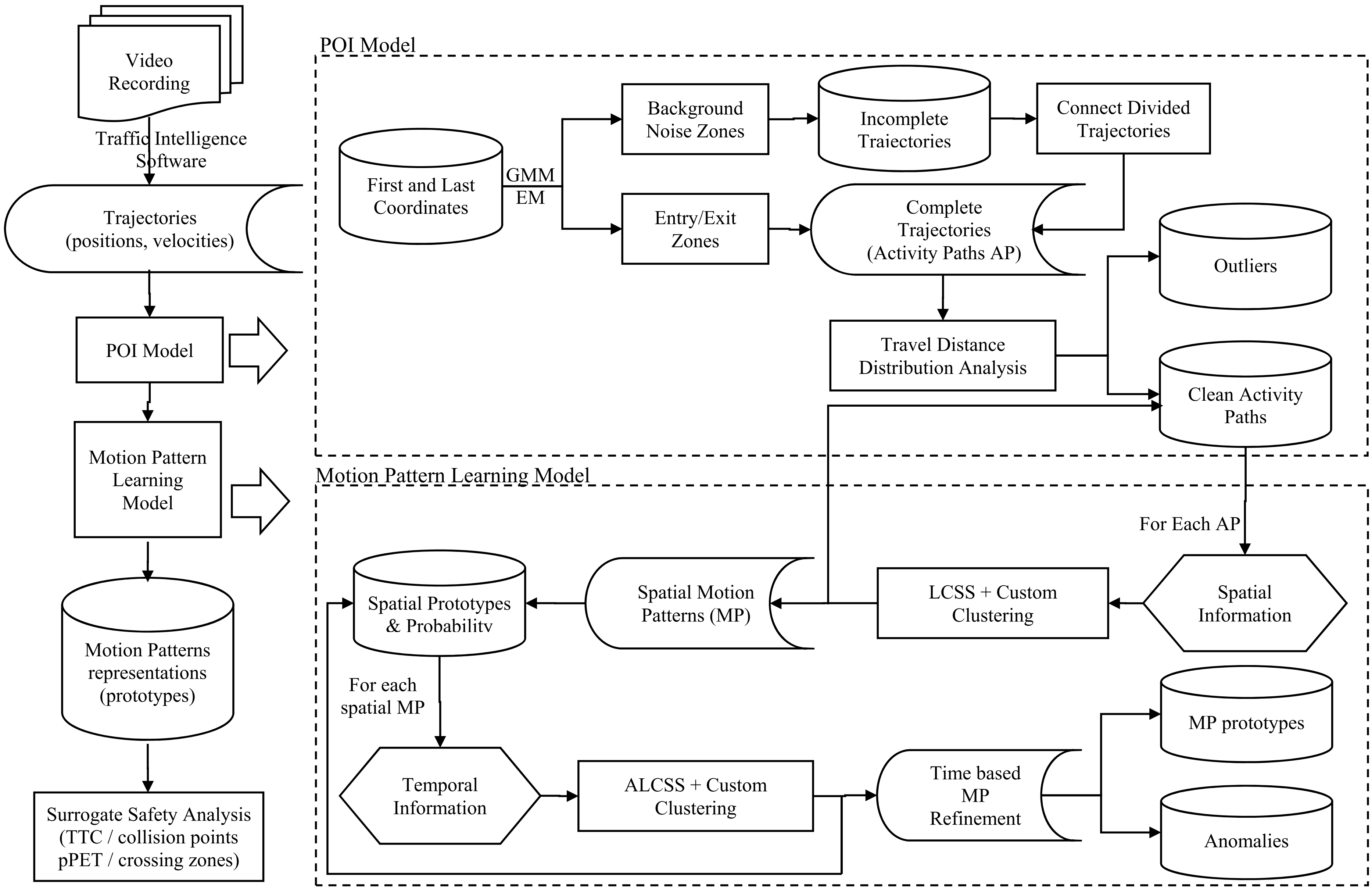
METHODOLOGY

Framework Overview

The multi-level framework is summarized in **FIGURE 1.** Road user behavior is learnt through two main models based on the road user trajectories:

1. **Points of interest (POI)**: as in [10], the POIs are learnt from trajectory endpoints based on GMM and EM algorithm and the results are entry and exit zones as well as clusters of noisy points (e.g. caused by moving occlusions and stopping/starting vehicles). Trajectories are complete if they connect an entry zone to an exit zone, which constitutes an activity path (AP).
2. **Motion patterns (MP)**: trajectories in each AP form the training dataset for MP learning using a two-stage trajectory clustering method based on spatial and temporal information. The MPs learnt for each AP at the first level are further clustered in the second stage. Each MP is represented by its longest trajectory, known as a prototype, and its associated probability is estimated using the size of each MP cluster. Moreover, outliers are detected and removed to minimize errors in the subsequent analyses. The prototypes are used for two main applications: anomaly detection and motion prediction for surrogate measures of safety. Anomalies are defined as unusual behaviours in terms of position and speed (e.g. evasive actions such as aggressive acceleration / deceleration, and tracking errors). Motion prediction relies on matching partial trajectories to the learnt spatial prototypes to evaluate the potential for collision.

FIGURE 1: Multi-level motion learning framework



Obtaining Road User Trajectories

As a first step, a video tracking tool from the open source “Traffic Intelligence” (TI) project [5] is used to detect and track moving road users: the result, road user trajectories, is the main input of the proposed framework. A trajectory is defined as a sequence of positions, from which velocities are derived, which constitute 4-dimensional flow vectors (position (x,y) and velocity (v_x,v_y))at each instant. Feature-based video tracking involves two steps. First, all distinct points (features) that move (more than a defined distance) are tracked from frame to frame. The second step is to group feature trajectories with similar motion for each road user. A road user is therefore represented by a set of feature trajectories and deriving one overall trajectory, ideally corresponding to the centroid, is not easy. The current default solution implemented in TI is the mean of the feature positions at each frame: this average trajectory is noisy and only suitable for visualization purposes. That is why MPs were learnt previously from feature trajectories [18]. Although the features track a road user well and have little noise, they have other issues such as:

1. They are fragmented which affects the detection of the entry/exit zones.
2. Many features represent one road user which affects the estimation of counts along each MP used for probabilistic motion prediction and complicates the detection of anomalies.
3. The feature trajectories constitute a relatively large dataset which affects the time necessary to learn the MPs.

Therefore, learning MPs based on road user trajectories would have benefits if noise in their trajectories can be reduced.

Road User Trajectory Smoothing based on Feature Trajectories.

Learning MPs using average trajectories is challenging as the noise in their positions often exceeds what can be accommodated by most clustering algorithms. A first step may be to smooth those trajectories. Standard techniques such as kernel smoothing, splines, Kalman filters are used in the literature. This section presents a novel smoothing method based on feature trajectories to reconstruct the object trajectory: the resulting object trajectory has less noise and better reflects the road user dynamics.

Smoothing is performed in two steps: 1) find a single feature that is tracked during the entire existence of the object or use the longest feature and complete the missing positions from other features, 2) find the parameters that represent the relationship between the feature F_1 and the mean feature position O, as shown in **FIGURE 2**a), represented by the vector $\overrightarrow{F_1 O}$. Since road users are rigid objects (pedestrians are somewhat less rigid), the angle θ between the feature velocity and $\overrightarrow{F_1 O}$, and the distance F_1O should be constant over time. Smoothing is therefore performed by computing the median angle θ and distance F_1O over time, θ^* and F_1O^*, and computing at each instant t the new object position as the feature translated by a vector of angle θ^* with the feature velocity at t and norm F_1O^*. The same procedure is repeated for all features with a minimum length: the final smoothed object is the median of all “new objects”.

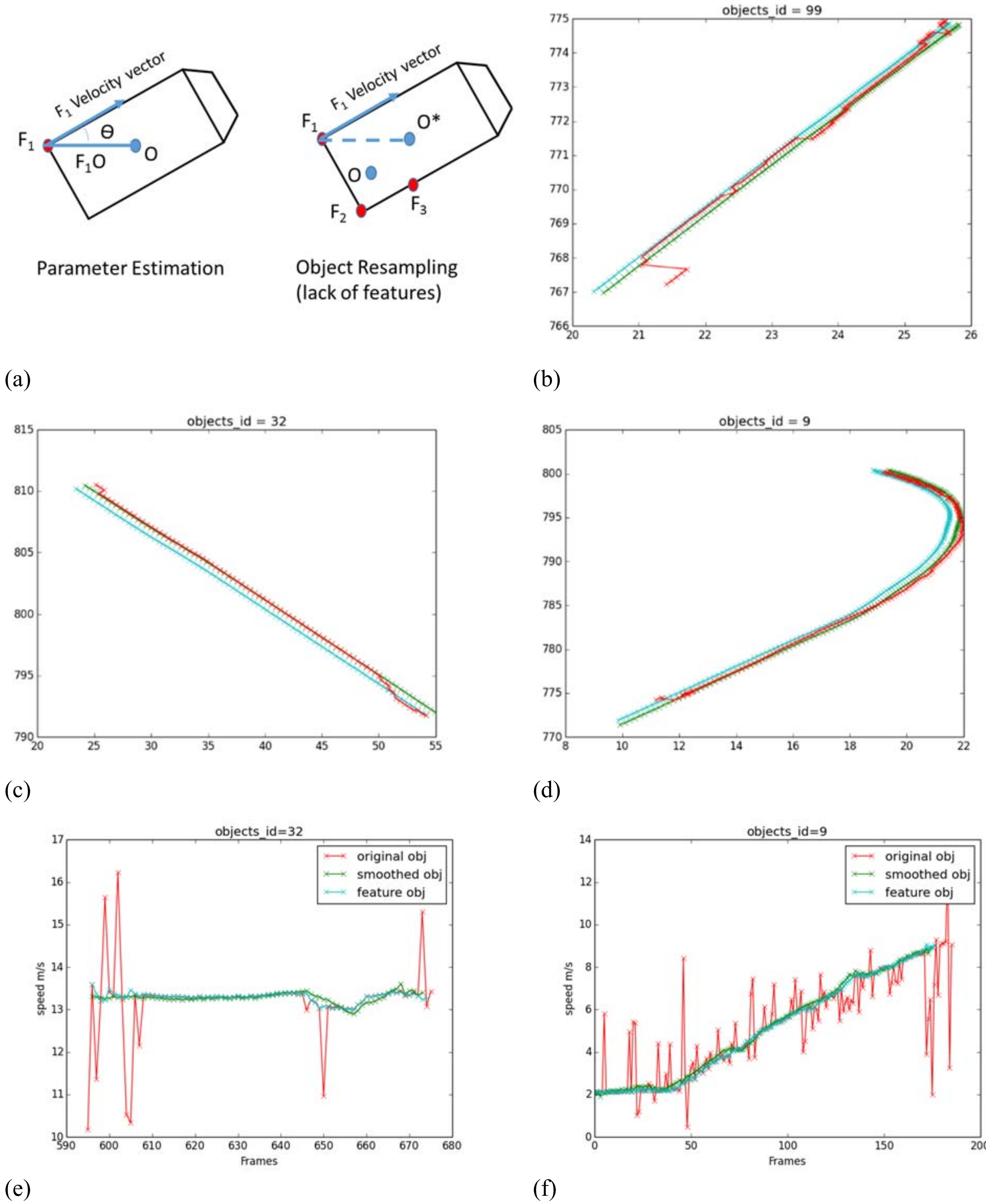


FIGURE 2: a) relationship between an object (vehicle) and one of its features; examples of smoothed road user trajectories (in green), with the corresponding original trajectories (in red) and the selected feature trajectory (in blue) are shown in b) for an overall noisy trajectory, c) for noise at the beginning and end, d) for a turning vehicle, (e,f) examples of derived speeds profile .

The smoothing algorithm performs well as shown in the examples of smoothed trajectories in **FIGURE 2**b-d. To further demonstrate the performance of the algorithm, the speed profiles derived from the original average road user trajectory (SP1), from the smoothed one (SP2), and from the average feature velocities provided by TI (SP3) are shown in **FIGURE 2**e,f: SP1 is very noisy, while SP2 is almost as smooth as SP3.

For a more quantitative measure of smoothness, the Cumulative Squared Jerk (CSJ), proposed in [23] as a cost function to plan the smoothest trajectory of a robotic arm, is computed for all trajectories based on SP1, SP2, and SP3. The CSJ is calculated for a two-dimensional trajectory (x(t),y(t)) using the following equation:

$$CSJ = \sum_{t_f}^{t_l} (\dddot{x}(t)^2 + \dddot{y}(t)^2)$$

Where $\dddot{x}$ and $\dddot{y}$ are the third time derivative of the x and y positions respectively, and t_f and t_l are the trajectory first and last instants.

For the two road users cases shown in **FIGURE 2**e, f, the CSJ values for SP2 (0.147 and 0.008) are much smaller than for SP1 (4.56 and 6.17) which confirms the effectiveness of the smoothing algorithm. Comparing the smoothness of SP2 and SP3, we found that SP3 is little smoother than SP2. Over the whole dataset, the mean and standard deviation (s.d.) of CSJ are largely reduced by the smoothing procedure as follows respectively for SP1, SP2 and SP3: 19.01 (s.d. 124.47), 0.565 (s.d. 4.77), and 0.075 (s.d. 0.18). SP3 is chosen for velocity computations since it is already computed.

POI Model

POI Detection

POIs are defined as zones where the road users appear, disappear, and/or stop for a specific moment. Traditionally, those zones are identified by some manually chosen polygons, which is error-prone and subjective, as well as tedious and time-consuming. Thus, identifying POIs automatically is recommended, modelling POIs with a GMM estimated using the EM algorithm as in [10].

The different types of POIs are estimated from different datasets specifically constructed. The entry and exit datasets are constituted respectively of the first and last positions of each road user trajectory. Notably, each dataset may contain points that do not correspond to actual road user entries to or exits from the scene (that mostly happens on the border of the image), but to areas with frequent tracking failures, caused by moving or static occlusions, i.e. occlusion by respectively a moving or static object such as a lamp or signal post. While static occlusion zones are similar to actual entry and exit zones, the moving occlusion zones are represented by wide Gaussian distributions, with lower density than the entry and exit zones. The algorithm steps for each point dataset are the following:

1) The EM algorithm is applied to the entry or exit dataset to estimate a GMM.

2) High-density Gaussian clusters correspond to entry and exit zones (and static occlusion zones) while low-density clusters correspond to zones of moving occlusion. Each cluster is classified according to its density d_i:

$$d_i = \frac{w_i}{\pi\sqrt{||\Sigma_i||}}$$

where w_i is the prior probability of Gaussian distribution i and $\sum_i$ is its covariance matrix. The classification threshold *Th* is defined as follow:

$$Th = \frac{\alpha}{\pi\sqrt{||\Sigma||}}$$

Where $\sum$ is the covariance matrix of the whole training dataset and α is a user-defined weight. The POIs that have low density, below *Th*, are classified as moving occlusion zones, often spread over most of the scene, akin to "background noise". Entry and exit zones will otherwise contain some static occlusion zones since they share the same density characteristics: in the current approach, static occlusion zones are identified manually.

POI Applications

Activity Paths and Trajectory Filtering The first application of POIs is to define APs by all trajectories moving from an entry to an exit zone. All trajectories that begin and end at an entry and exit zone is a complete trajectory, while the rest are incomplete trajectories.

The dataset of complete trajectories is used in the MP learning. Because outliers may remain in this dataset, a pre-processing procedure is recommended. The proposed approach detects outliers by analyzing the distribution of the travelled distance in each AP: it is expected that vehicles moving along a given AP will have similar travelled distances. The extreme distances therefore represent outliers, which are identified using boxplots and their traditional statistics: the median, the first (Q1) and third (Q3) quartile, the interquartile range (IQR=Q3-Q1) and the "whisker" limits typically defined as Q1-1.5 IQR and Q3+1.5 IQR. The usual application is to consider points outside of the whiskers as outliers: for this application, a further distinction is made for outliers beyond Q1-3 IQR and Q3+3 IQR which correspond to grouping or smoothing errors and are therefore removed from the AP, and other "mild" outliers corresponding mostly to lane changes which are kept. These outliers can be reviewed for a better understanding of the scene and activities.

At the end of this stage, three trajectory datasets are constituted: clean complete trajectories, incomplete trajectories, and outlier complete trajectories. The second dataset can be analyzed to merge incomplete trajectories into complete ones, then the first dataset will be used for the MP algorithm.

Connecting Incomplete Trajectories Connecting incomplete trajectories is performed by logical connection procedures: if a trajectory does not end in an exit zone, all trajectories that start within a defined distance from the end are identified. Then, any two trajectories moving in the same direction (tested by the velocity angle) with small enough time difference may be merged. To avoid searching through the whole dataset, incomplete trajectories that start or end within each detected static occluded zones are first identified, and the ones satisfying the logical constraint are merged. The rest of the dataset is finally processed systematically, considering all incomplete trajectories not ending in an exit zone with incomplete trajectories not starting in an entry zone. This simple procedure is useful to speed up the identification of incomplete trajectory connection.

Efficiency Gains for MP Learning and Motion Prediction The most common challenge of learning MPs, i.e. of clustering trajectories, is to compute the similarities of all pairs of trajectories. To avoid that, the trajectory dataset is divided into different subsets corresponding to APs. Learning the MPs for each AP separately reduces significantly the computation cost and required space.

Regarding motion prediction, a road user enters the scene in a known entry zone and its partial trajectory (at each instant) needs to be matched only to the MP prototypes that share the same entry zone without the need to compute the similarity to all the MP prototypes in the scene. This simple procedure is efficient to speed up motion prediction. Besides, because the road user usually has a destination in mind, at least at the typical scale of the zones of study, one should not predict that it may leave the scene through another exit than the one he actually took. Therefore, the partial trajectory is matched only the prototypes with the same entry and exit zones.

Two-Stage MP Learning

In the proposed approach, a slight variation of the algorithm previously developed to cluster MPs[18] is implemented for both stages. However, two different similarity measures are used for each stage as presented in the following sub-sections.

Spatial MP Similarity Measure

The spatial information is constituted by the position coordinates in each trajectory. With the purpose of comparing the trajectories without pre-processing that would distort the data, a similarity measure should be able to handle variable length inputs. LCSS can deal with variable length vectors and is robust to noise and outliers as some points may not be matched.

The LCSS definition is taken from [24]. Let A and B be a two trajectories with size n and m respectively, where A = $\{a_1, a_2, \ldots a_n\}$, B= $\{b_1,b_2,\ldots,b_m\}$, and a_i and b_j are the object position coordinates. For a trajectory A, let Head (A) be the sequence $\{a_1,a_2,\ldots,a_{n-1}\}$. Given a distance function d (the Euclidean distance is used) and a matching threshold ε, the LCSS between two time series A and B can be calculated iteratively as follows:

$$LCSS(A,B) = \begin{cases} 0 \quad if\ A\ or\ B\ is\ empty \\ 1 + LCSS\big(Head(A), Head(B)\big), \quad if\ d(a_n, b_m) < \epsilon \\ \max\Big(LCSS(Head(A), B), LCSS\big(A, Head(B)\big)\Big), otherwise \end{cases}$$

In addition, a parameter (δ) can be added to control how far in time it can go in order to match a point in a trajectory to a point in another trajectory, known as trajectory bounds (positions a_i and b_j are compared only if $|i-j|\leq\delta$). Moreover, to be independent of trajectory length, the LCSS is divided by the minimum lengths to yield a similarity measure (SLCSS) between the two trajectories (A, B) and a distance measure (DLCSS) defined as follows:

$$SLCSS\ (A,B) = \frac{LCSS(A,B)}{\min(n,m)}$$

$$DLCSS\ (A,B) = 1 - SLCSS\ (A,B)$$

Temporal MP Similarity Measure

The first stage of MP learning makes use only of spatial information. Temporal dynamics, measured in particular by speed, are important motion characteristics that may vary within a spatial activity path represented by the same prototype trajectory learnt in the first stage. Therefore, the speed profile should be studied for each cluster generated by the first stage. To differentiate properly speed profiles, a similarity measure that considers the rate of change of profiles is needed. The Aligned Longest Common Sub-sequence (ALCS) proposed in the authors' previous work [4] is used for this stage.

ALCS was developed after the observation that the existing formulations of the LCSS, with or without δ, are insufficient to measure the similarity of series if the series are simply shifted with respect to each other or if series with different rates of change should be considered different. ALCS finds the best alignment of two series while taking into account a finite δ, allowing taking into account the rates of change. The length of the aligned longest common sub-sequence is computed by simply shifting the two series with respect to each other, i.e. by adding an integer parameter *shift* to the *LCSS* computation (replacing the condition $|i-j|\leq\delta$ by $|i-shift-j|\leq\delta$) and taking the maximum *LCSS* for all possible *shift* values. The corresponding aligned similarity measure *ALCSS* and distance *DALCS* are defined similarly.

The Clustering Method
Choosing a type of cluster representation and a similarity measure rules out some clustering methods. The choice of processing the trajectories and speed profiles in their original shape with variable lengths rules out for example classical clustering algorithms such as k-means since the concept of a centroid is not defined.

The algorithm used for the results presented in this paper is the same as [4] and is a slight variation of the algorithm previously developed to cluster motion patterns [18]. The differences from the previous work are twofold: the first is to sort the elements (trajectories or speed profiles) according to their length, to start considering first the longest elements and the second is to keep the longer prototype when two clusters are merged. The algorithm parameter is the minimum similarity for two elements to be in the same cluster: when learning prototypes, an element will be added as a new prototype if its maximum similarity to all existing prototypes is lower than the parameter. Unlike other clustering algorithms, it is not necessary to compute the similarities between all elements: it is only necessary for each trajectory to compute its similarity to all existing prototypes at the current stage of the algorithm. This also speeds up the clustering. Another variation is to avoid clusters with few assigned trajectories. Thus, a minimum cluster size is used: if a cluster contains fewer elements than the defined minimum size, its prototype is removed from the set of prototypes and the associated trajectories are assigned to the most similar prototype. These steps are called the prototype refinement algorithm.

MP Applications: Motion Prediction and Surrogate Safety Indicators
Given a set of prototypes and their matching counts (their cluster size), future motion can be predicted with some probability. Let T be a trajectory of a road user entering the scene and T_* the partial trajectory to the current frame. The partial trajectory will be compared with all spatial prototypes with the same entry zone and future positions are predicted based on the temporal prototypes associated with the matched spatial prototype.

Only prototypes ϕ_i whose similarity exceeds a defined threshold are considered as a future trajectory. The probability of following each prototype is estimated using the Bayes rules:

$$P(\phi_i|T_*) = \frac{P(T_*|\phi_i)\ P(\phi_i)}{\sum_{n=1}^{N} P(T_*|\phi_n)\ P(\phi_n)}\ , \qquad i = 1, 2, \dots N$$

Where $P(T_*|\phi_i)$ is represented by DLCSS as a weighted factor, $P(\phi_i)$ is estimated by the ratio of trajectories corresponding to ϕ_i to the number of trajectories on all matched prototype MPs.

For each pair of road users in an interaction (close enough) at each instant, two safety indicators are computed using the predicted trajectories: for each pair of predicted trajectories for the two road users, if the trajectories intersect at the same time, there is a predicted collision point (CP) and the time of arrival is the TTC, otherwise if they only intersect, the time difference at the crossing zone (CZ) is the predicted Post Encroachment Time (pPET). The final TTC and/or pPET at each instant are the weighted averages over respectively all CPs and CZs. Readers are referred to [1] for more details.

EXPERIMENTAL STUDY

The proposed approach is evaluated on a case study from a large dataset of video recordings. The case study is the intersection of Guy Street and Boulevard Rene Levesque in downtown Montreal, Canada:

video data is captured using a consumer camera at a resolution of 1280 x 720 pixels (at 29.97 fps) and were recorded for around one hour from a high rise building facing the intersection. The above algorithms are implemented in the open source Python language using several scientific libraries, in particular scikit-learn (available at http://scikit-learn.org/stable/index.html) for GMM and EM, and most are or will be available in the Traffic Intelligence project.

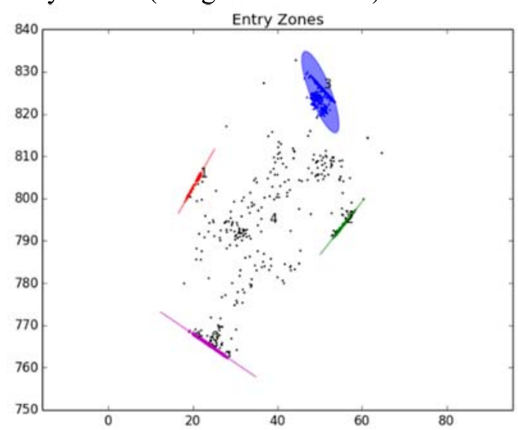
For this analysis, only vehicle trajectories are used to detect the POI zones. Road users are classified based on speed information, maximum speed over time in this case with a threshold of 15 km/h. The parameter for GMM learning is the number of components or expected zones in the scene, including the noise clusters. Although components numbers can be estimated automatically using Bayesian Information Criteria (BIC), it suffers in most cases from over fitting (selecting more clusters than necessary). Therefore, the number of components is found by trial and error for each scene.

POIs were learnt and classified into entry zones, exit zones, and noise clusters based on the density criterion with a defined weight α equal to 1.0. The results shown in **FIGURE 3**a-d used a number of five and six components to cluster entry and exit datasets respectively. They include the four entry and the four exit zones. In addition, the static occluded zone under the pole in the top-right corner of the scene is detected as an extra exit zone, which was however not detected in the entry zones. This is due to the occluded zone closeness to the entry zone, so that it is merged with the closest entry zone producing a relatively wider Gaussian distribution. In this dataset, tracking failures caused by moving occlusion are clustered as a large Gaussian noise cluster over the whole scene. **FIGURE 3**a) and b) represent the entry and exit zones including the occluded zone overlaid over a camera image. All detected POIs and their covariance ellipses are shown in **FIGURE 3**c,d) in world coordinates.

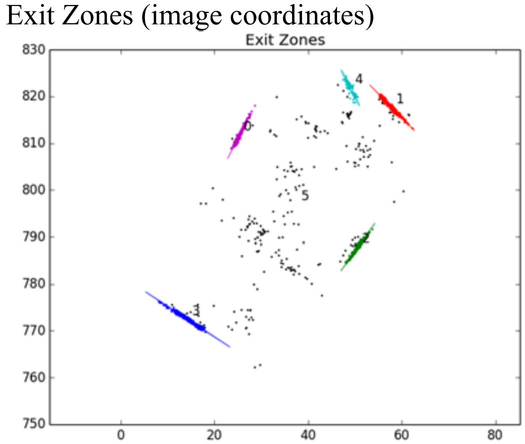
Using the detected zones, trajectories are filtered into complete and incomplete trajectories. The complete trajectories are shown in **FIGURE 3**e) before filtering for outliers. This is done based on the travelled distance distribution that can be seen as boxplots in **FIGURE 3**f). The points beyond the whiskers are outliers (examples shown in **FIGURE 3**g). For this case study, it was not possible to reconnect incomplete trajectories without adding errors to the dataset of complete trajectories and the procedure was therefore not applied.

a) Entry Zones (image coordinates)

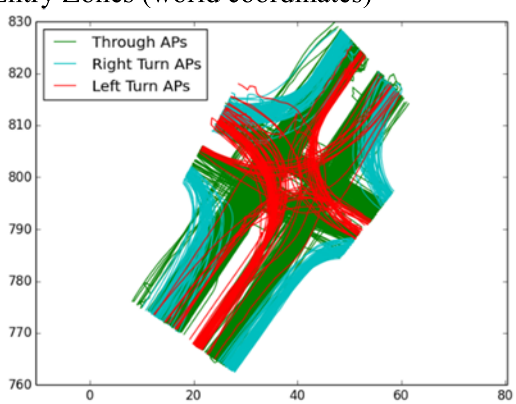
b) Exit Zones (image coordinates)



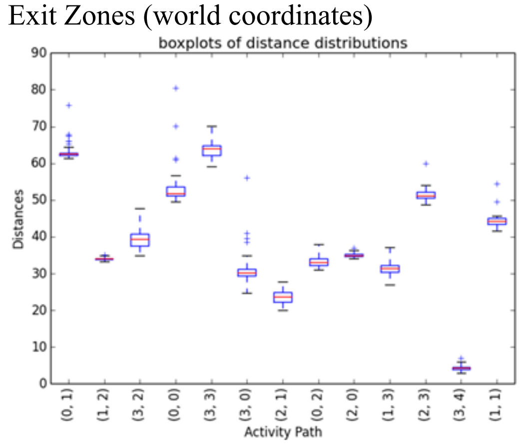
c) Entry Zones (world coordinates)



d) Exit Zones (world coordinates)



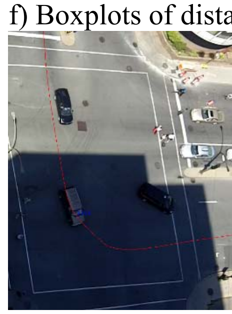
e) Complete Trajectories (APs)



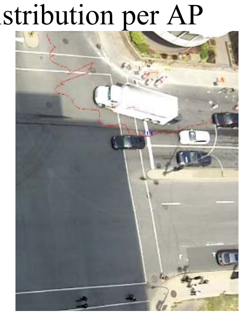
f) Boxplots of distance distribution per AP

aggressive lane change

abnormal overtaking



abnormal left turn



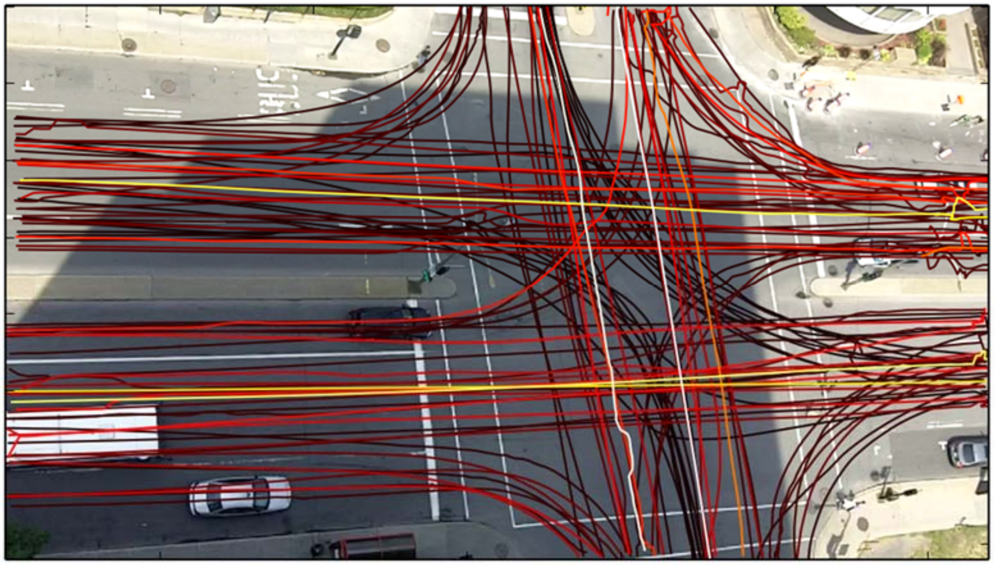
over-grouping with smoothing errors

g) Examples of detected outliers based on distance distribution

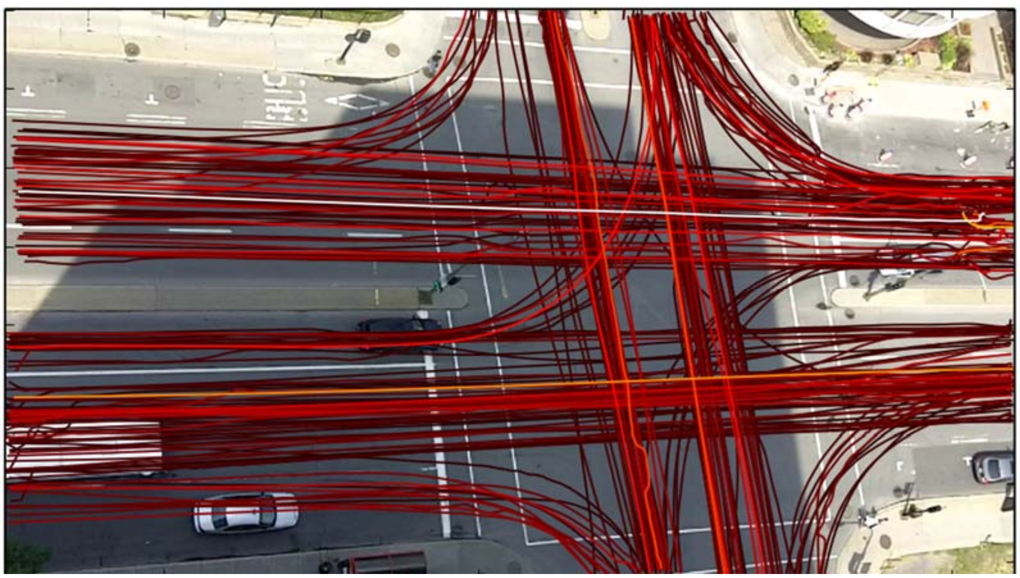
FIGURE 3: POI learning and their applications

For MP learning, the dataset contains 1379 complete trajectories after removing the outliers. The parameters for learning phase are the matching threshold ε and the minimum cluster similarity. These were chosen by trial and error respectively as 1.5 m and 0.75 for spatial information and 1 m/s and 0.75 for temporal information. The learnt prototypes for the two learning stages are presented in **FIGURE 4**a,b). Although such unsupervised learning is difficult to evaluate, the results of MP learning based on spatial information suggest an acceptable division of the trajectories (see **FIGURE 4**a). As expected, results based on temporal information provide more prototype trajectories. After MPs are learnt, trajectories that are not assigned to any MP are considered as anomalies. The parameter for anomaly detection is the minimum cluster size that was selected as the largest of 3 trajectories or 10 % of the MP size for spatial and temporal information. The detected anomalies should be investigated carefully by analyzing them manually. In our dataset there are many sources of anomalies such as road user misclassification, tracking issues, normal but rare movements, and safety problems such as changing lane by crossing a white solid line which could indicate a side-swipe interaction with or without collision and excessive speed. Samples of spatial and temporal anomalies for the studied intersection are summarized in **FIGURE 4**c).

Our objective is to apply the learnt MPs for surrogate safety analysis. Figure 5a,b) presents an example of a Left Turn and Opposite Direction (LTOD) interaction at the instant where both vehicles enter the intersection central area (at instant 80490): motion prediction is made based on MP and Constant Velocity (CV) at each instant of interaction over a time horizon of 150 frames (5 seconds). Using MPs, the predicted speed profiles for through movement are shown in Figure 5c). Accordingly, safety indicators are computed based on MPs and CV and plotted in Figure 5d). Notably, the proposed algorithm predicts indicator values earlier than the method based on CV (around 2 sec) and even before the left turn vehicle enters the physical area. Some TTC values are similar in both methods when the two vehicles are very close to the observed trajectory intersection in which case possible predicted trajectories are basically limited to CV. On the contrary, pPET values are further apart, especially before the vehicles reach the trajectory intersection, because of the unrealistic CV-based motion prediction for the left turn movement. In our algorithm, the locations of estimated CPs and CZs are concentrated around the trajectory intersection as expected. However, the CV-based CZs are shifted away from the trajectory intersection.



a) First-stage (position-based) prototypes (129 prototypes)

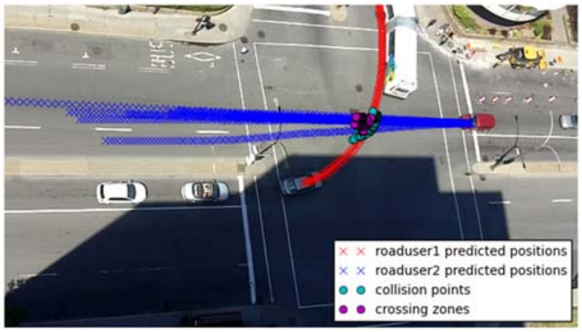


b) Second-stage (speed-based) prototypes (259 prototypes)

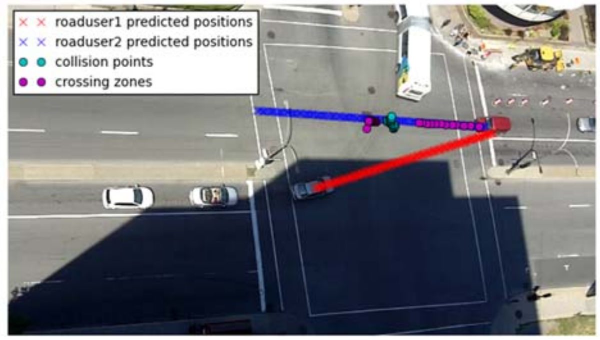
(i) (ii) (iii) (iv) (v) (vi)

c) Samples of detected anomalies: i) misclassification (cyclist), ii) misclassification (pedestrian jogging), iii) over-grouping problem, iv) normal movement that rarely happens (bus change lane to its exclusive lane), v) lane change over solid lane markings with no surrounding vehicle, and vi) side-swipe interactions (one vehicle changes lane over a solid lane marking while another vehicle adapts its path at the same time)

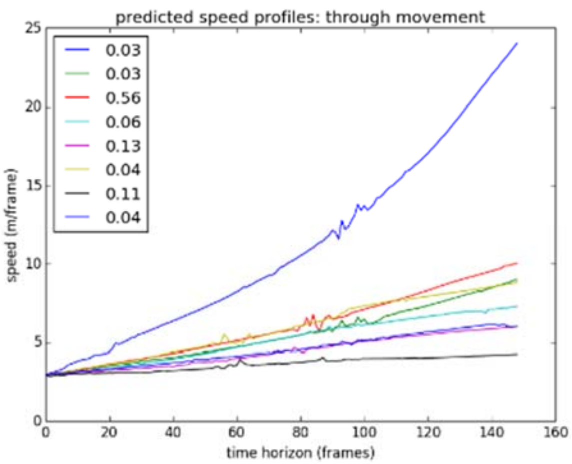
FIGURE 4: Prototypes representing MPs (color-coded from dark red to yellow to represent the number of trajectories associated to the MP) and samples of anomaly detections



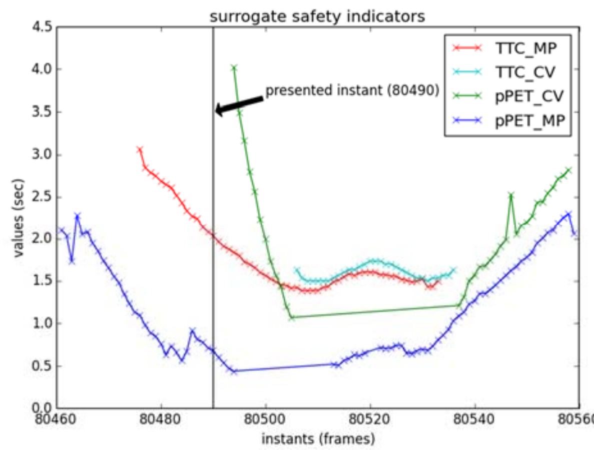
a) MP based motion prediction at instant 80490



b) CV based motion prediction at instant 80490



c) Predicted speed profiles for the through movement and its associated probability



d) Surrogate safety indicators based on MP and CV methods

FIGURE 5: Examples of predicted trajectories (the blue and red crosses or lines in the top graphics) and the corresponding plots of TTC and pPET for an LTOD interaction.

To test the effectiveness of our algorithm in speeding up the computation for MP learning, a sample of 200 complete trajectories is used to construct their complete similarity matrix based on the traditional method without dividing the dataset first into sub-datasets (APs). The run time of the traditional method is 1600 seconds, while the proposed method takes only 140 seconds: the gain is more than 90 %. Furthermore, for motion prediction, a sample of 100 partial trajectories is matched to all prototypes. This (traditional) approach takes 1240 seconds, while the new approach with AP logical constraints takes only 200 seconds: the gain is also very high, around 85 %, while providing the same results.

CONCLUSION

This paper introduced an effective and generic framework for understanding driver behaviour in an unsupervised manner that can be applied to surrogate safety analysis. New algorithms for multi-level MP learning have been proposed that enable automated scene interpretation, anomalous behaviour detection and surrogate safety analysis. The first level is to detect POIs that are then used to enhance the MP learning and behaviour prediction. This was performed in two ways: by removing the outliers and proposing logical constrains to decrease the processing time. The second level is MP learning, which studies spatial information of trajectories to produce set of prototypes, and then integrates speed profiles for each spatial MP. The resulting prototypes are used for motion prediction. An intersection case study demonstrates the ability of the proposed approach to reduce the computation cost up to 90 % for MP learning, to clean the trajectory dataset from tracking outliers, to use actual trajectories as prototypes without any pre- or post-processing and to predict realistic future motion based on spatial and temporal (speed) information to compute surrogate safety indicators (e.g. TTC, pPET). Comparing the surrogate safety indicators based on MPs with the traditional constant velocity method on a LTOD conflict case, our algorithm is able to compute the surrogate indictors early with a larger number of measurements. Moreover, collision points and crossing zones were located around the intersection of the observed vehicle trajectories. This finding adds further support to the findings in [1] that constant velocity is not a realistic method to compute surrogate indicators and to understand the collision processes. Future work will refine the different parts of the framework, including the method to reconnect disrupted road user trajectories. The framework will also be applied to several other case studies and to other types of road users, in particular vulnerable ones. In addition, different motion prediction methods that are developed in previous work [1] and this framework should be compared for surrogate safety analysis.

ACKNOWLEDGEMENT

The authors wish to acknowledge the financial support of the Natural Sciences and Research Council of Canada (NSERC).

REFERENCES

[1] M. Mohamed and N. Saunier, "Motion Prediction Methods for Surrogate Safety Analysis," *Transportation Research Record: Journal of the Transportation Research Board,* vol. 2386, no. -1, pp. 168-178, December 2013.

[2] N. Saunier, T. Sayed and K. Ismail, "Large Scale Automated Analysis of Vehicle Interactions and Collisions," *Transportation Research Record,* vol. 2147, pp. 42-50, 2010.

[3] C. Hydén and F. H. Amundsen, "Proceedings: first workshop on traffic conflicts, Oslo 77.," 1977.

[4] N. Saunier and M. G. Mohamed, "Clustering Surrogate Safety Indicators to Understand Collision Processes," in *Transportation Research Board, 93rd meeting*, Washington,USA, 2014.

[5] N. Saunier, "Traffic Intelligence software," 2014. [Online]. Available: https://bitbucket.org/Nicolas/trafficintelligence. [Accessed 26 07 2012].

[6] W. Hu, T. Tan, L. Wang and S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors," *IEEE Transactions on Systems, Man, and Cybernetics - Part C,* vol. 34, no. 3, pp. 334-352, August 2004.

[7] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 18, no. 8, pp. 1114-1127, 2008.

[8] C. Stauffer, "Estimating Tracking Sources and Sinks," in *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03. Conference on*, 2003.

[9] S. McKenna and H. Nait-Charif, "Learning spatial context from tracking using penalised likelihoods," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2004.

[10] D. Makris and T. Ellis, "Learning semantic scene models from observing activity in visual surveillance," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 35, no. 3, pp. 397-408, June 2005.

[11] X. Wang, K. Tieu and E. Grimson, "Learning Semantic Scene Models by Trajectory Analysis," in *In ECCV (3) (2006*, 2006.

[12] B. Morris and M. Trivedi, "Trajectory Learning for Activity Understanding: Unsupervised, Multilevel, and Long-Term Adaptive Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 33, no. 11, pp. 2287-2301, 2011.

[13] M. Nedrich and J. Davis, "Detecting behavioral zones in local and global camera views," *Machine Vision and Applications,* vol. 24, no. 3, pp. 579-605, 2013.

[14] M. Bennewitz, W. Burgard, G. Cielniak and S. Thrun, "Learning motion patterns of people for compliant robot motion," *Internationl Journal of Robotics Research,* vol. 24, pp. 31-48, 2005.

[15] W. Hu, X. Xiao, D. Xie and T. Tan, "Traffic Accident Prediction Using 3-D Model-Based Vehicle Tracking," in *IEEE Transactions on Vehicular Technology*, 2004.

[16] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan and S. Maybank, "A system for learning statistical motion patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 28, no. 9, pp. 1450-1464, Sept 2006.

[17] S. Atev, O. Masoud and N. Papanikolopoulos, "Learning Traffic Patterns at Intersections by Spectral Clustering of Motion Trajectories," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006.

[18] N. Saunier, T. Sayed and C. Lim, "Probabilistic Collision Prediction for Vision-Based Automated Road Safety Analysis," in *The 10th International IEEE Conference on Intelligent Transportation Systems*, Seattle, 2007.

[19] D. Parkhurst, "Using digital video analysis to monitor driver behaviour at intersections," Center for Transportation Research and Education, Iowa State University, 2006.

[20] A. Laureshyn, K. Åström and K. Brundell-Freij, "From Speed Profiles Data to Analysis of Behaviour: Classifictation by Pattern Recognition Technology," in *IATSS Research 33 (2)*, 2009.

[21] X. Wang, K. T. Ma, G.-W. Ng and W. E. L. Grimson, "Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models," *International journal of computer vision,* vol. 95, no. 3, pp. 287-312, 2011.

[22] R. Zhao and X. Wang, "Counting Vehicles from Semantic Regions," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 14, no. 2, pp. 1016-1022, 2013.

[23] T. Flash and N. Hogans, "The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model," *Journal of neuroscience,* vol. 5, pp. 1688-1703, 1985.

[24] M. Vlachos, G. Kollios and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proceedings of the 18th IEEE International Conference on Data Engineering*, 2002.