Unsupervised Trajectory Modeling based on Discrete Descriptors for Classifying Moving Objects in Video Sequences

Damián Campo
Mohamad Baydoun
Lucio Marcenaro
Andrea Cavallaro
Carlo Regazzoni

October 2018

Queen Mary
University of London

UNIVERSITÀ DEGLI STUDI
DI GENOVA

ISIP40

CONTENT

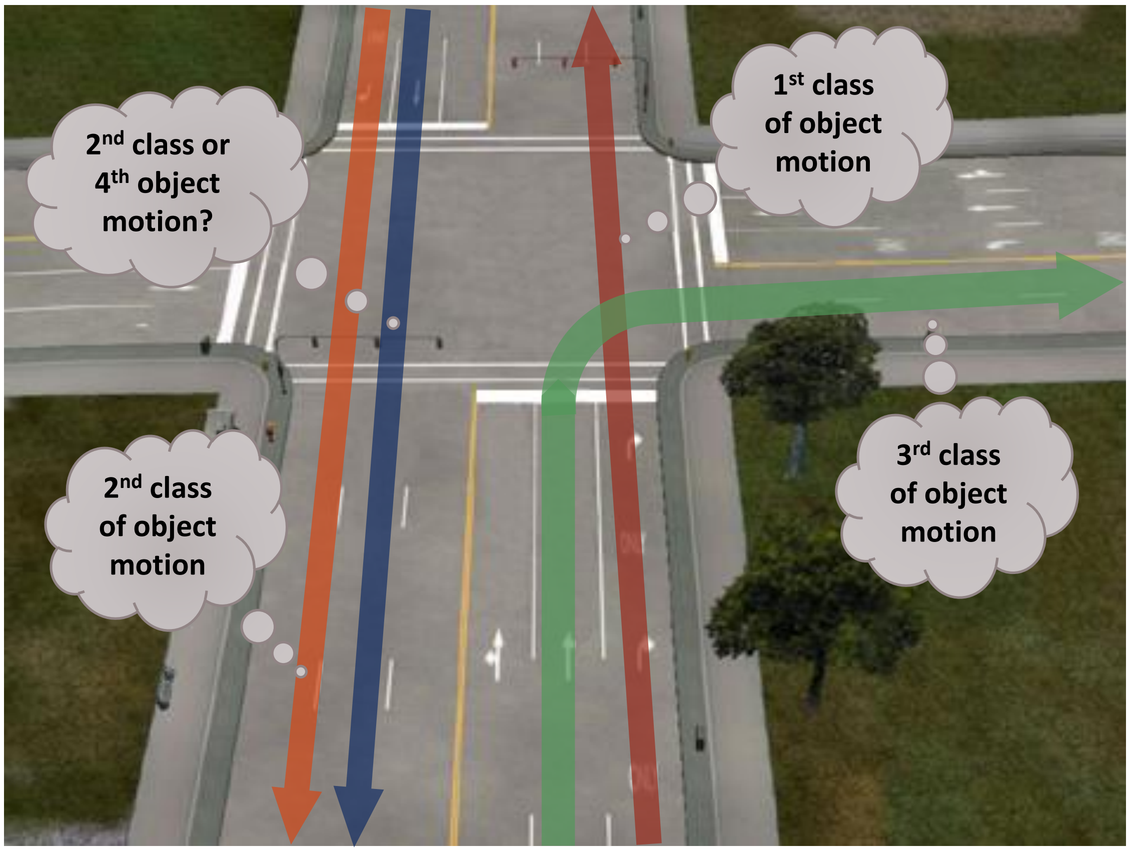
1. Introduction to the main problem
2. Proposed method

2.1. Offline vocabulary learning

2.2. Incremental classification process

3. Employed datasets and experimental results
4. Conclusions and future work

Introduction to the problem



- Is it possible to have a set of descriptors that encode observed motions?
- Is it possible to distinguish trajectories with different dynamics appearing in the same location?
- Is it possible to classify the observed trajectories incrementally, i.e., as observations arrive?

Proposed method

Video frame in timestamp k

For a single object

$$X_k = \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \\ t_k \end{bmatrix}$$

Location in the scene (2-dimensional)

Velocity information (2-dimensional)

Time spent in the video sequence (1-dimensional)

Clustering of similar state information

Video sequence

State sequences of objects

Identified objects ↓

$$
\begin{array}{ccccc}
X_k^1 & X_{k+1}^1 & \cdots & X_{K_1-1}^1 & X_{K_1}^1 \\
X_k^2 & X_{k+1}^2 & \cdots & X_{K_2-1}^2 & X_{K_2}^2 \\
\vdots & \vdots & \cdots & \vdots & \vdots \\
X_k^{N_k} & X_{k+1}^{N_{k+1}} & \cdots & X_{K_{N_{K-1}}-1}^{N_{K-1}} & X_{K_{N_K}}^{N_K}
\end{array}
$$

Trajectories →

State sequences of objects → **Self organizing map (SOM) training** → **Clusters of states (vocabulary)**

$$\omega_{SOM} = [\beta, \alpha, \gamma]$$

$\beta + \alpha + \gamma = 1$
β: Location weight
α: Velocity weight
γ: Spent time weight

$$\boldsymbol{C} = \{C_1, C_2, \cdots, C_M\}$$

5-dimensional regions encoding objects' dynamics

Vocabulary properties

Vocabulary

$$\boldsymbol{C} = \{C_1, C_2, \cdots , C_M\}$$

5-dimensional regions encoding objects' dynamics

Letters

$$C_m = \begin{bmatrix} \bar{x}_m \\ \bar{y}_m \\ \bar{\dot{x}}_m \\ \bar{\dot{y}}_m \\ \bar{t}_m \end{bmatrix}$$

Where $C_m \in \boldsymbol{C}$

Distance between letters

$$d_{i,j} = (\omega_{SOM} A)\ \mathrm{abs}\big(C_i - C_j\big).$$

Where:

$$\omega_{SOM} = [\beta, \alpha, \gamma] \ \ ; \ A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix};$$

$$C_i \in \boldsymbol{C} \ \text{ and } \ C_j \in \boldsymbol{C}$$

A **distance matrix** D containing the **separation between letters** is defined as:

$$D = \begin{bmatrix} 0 & \cdots & d_{i,j} \\ \vdots & \ddots & \vdots \\ d_{i,j} & \cdots & 0 \end{bmatrix}$$

Words generation

Any 5-dimensional state X_k can be **transformed into** a vocabulary **letter** by following the function $G(X_k)$, defined as:

$$G(X_k) = m_{X_k} = \arg\min_m\big((\omega_{SOM}\,A)\;abs(X_k - C_m)\big)$$

Let a trajectory l be defined as:

$$\tilde{\boldsymbol{X}}_l = \{X_1^l, \dots, X_{k_l}^l, \dots, X_{K_l}^l\}$$

$$G(\tilde{\boldsymbol{X}}_l)$$

Discrete version of trajectory l:

$$m_{\tilde{\boldsymbol{X}}_l} = \{m_{X_1^l}, \dots, m_{X_{k_l}^l}, \dots, m_{X_{K_l}^l}\}$$

"Word" (class) generation

Given a set of trajectories:

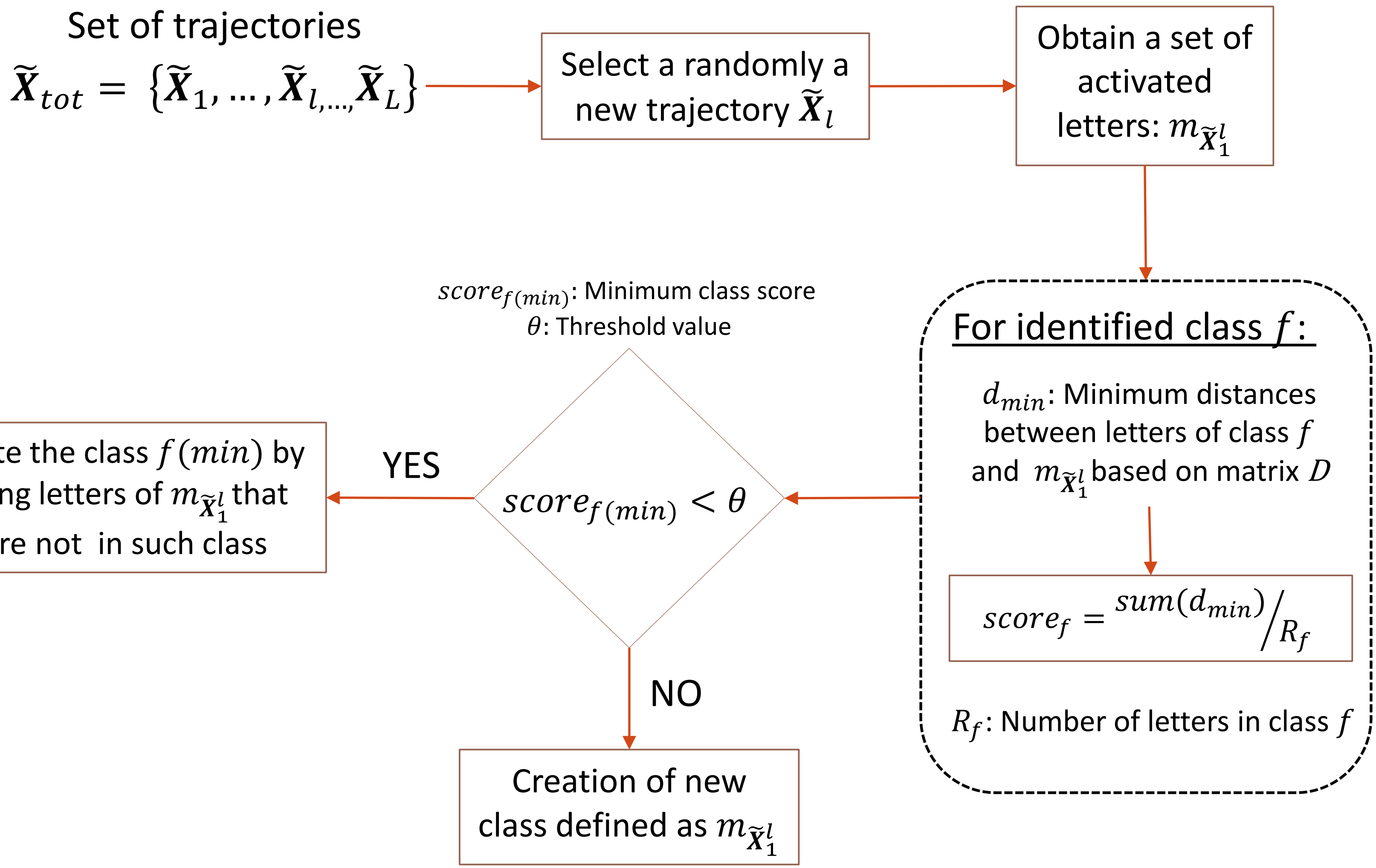
$$\tilde{\boldsymbol{X}}_{tot} = \{\tilde{\boldsymbol{X}}_1, \dots, \tilde{\boldsymbol{X}}_{l,\dots,}\tilde{\boldsymbol{X}}_L\}$$

How to generate a **dictionary** (**set of words**) that encodes observed dynamics for classification purposes?

Can such dictionary be built as **trajectories** are **observed one by one**?

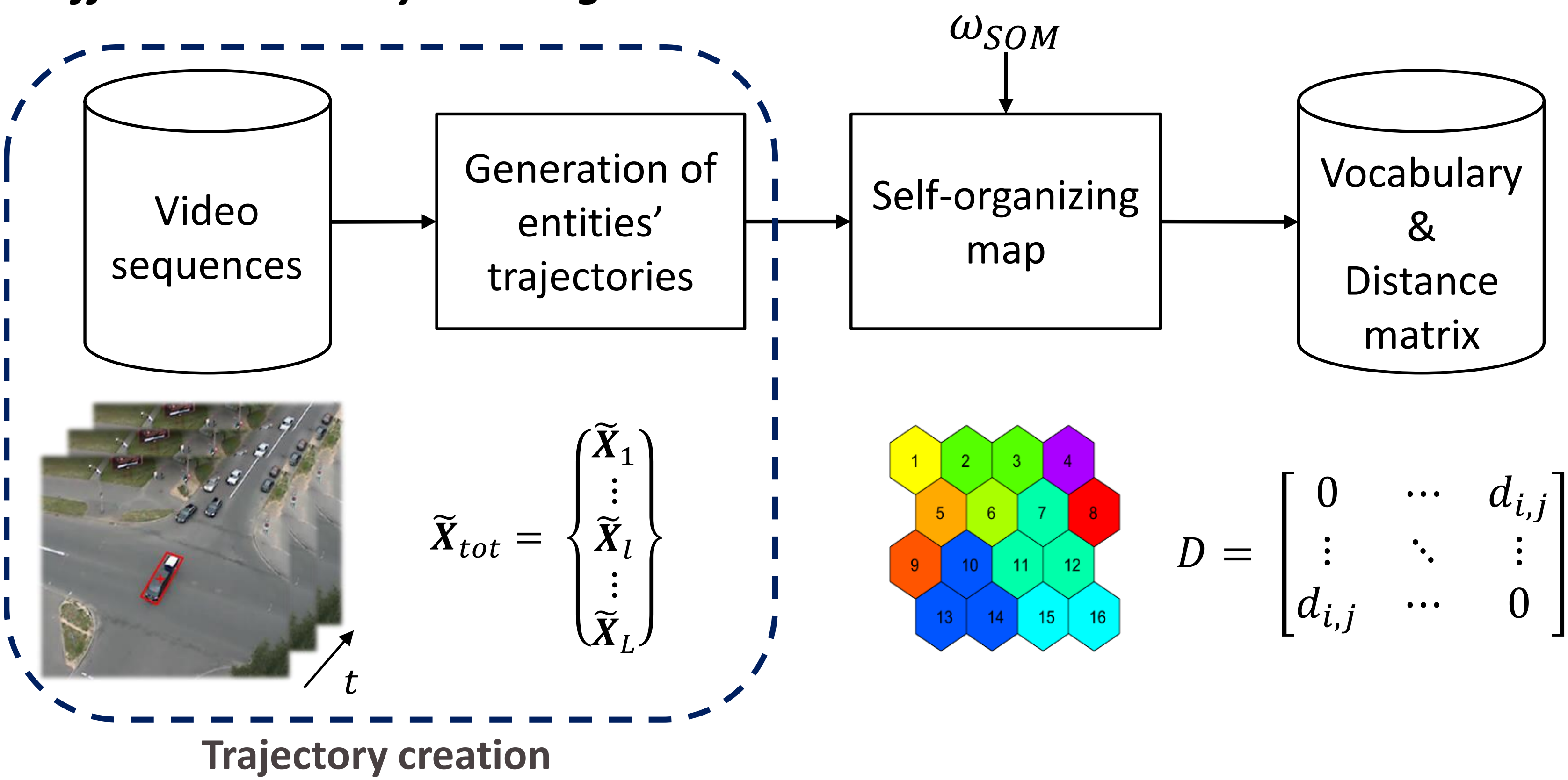
Can such dictionary be **quasi-invariant** to the **order of observed trajectories**?

Incremental dictionary creation

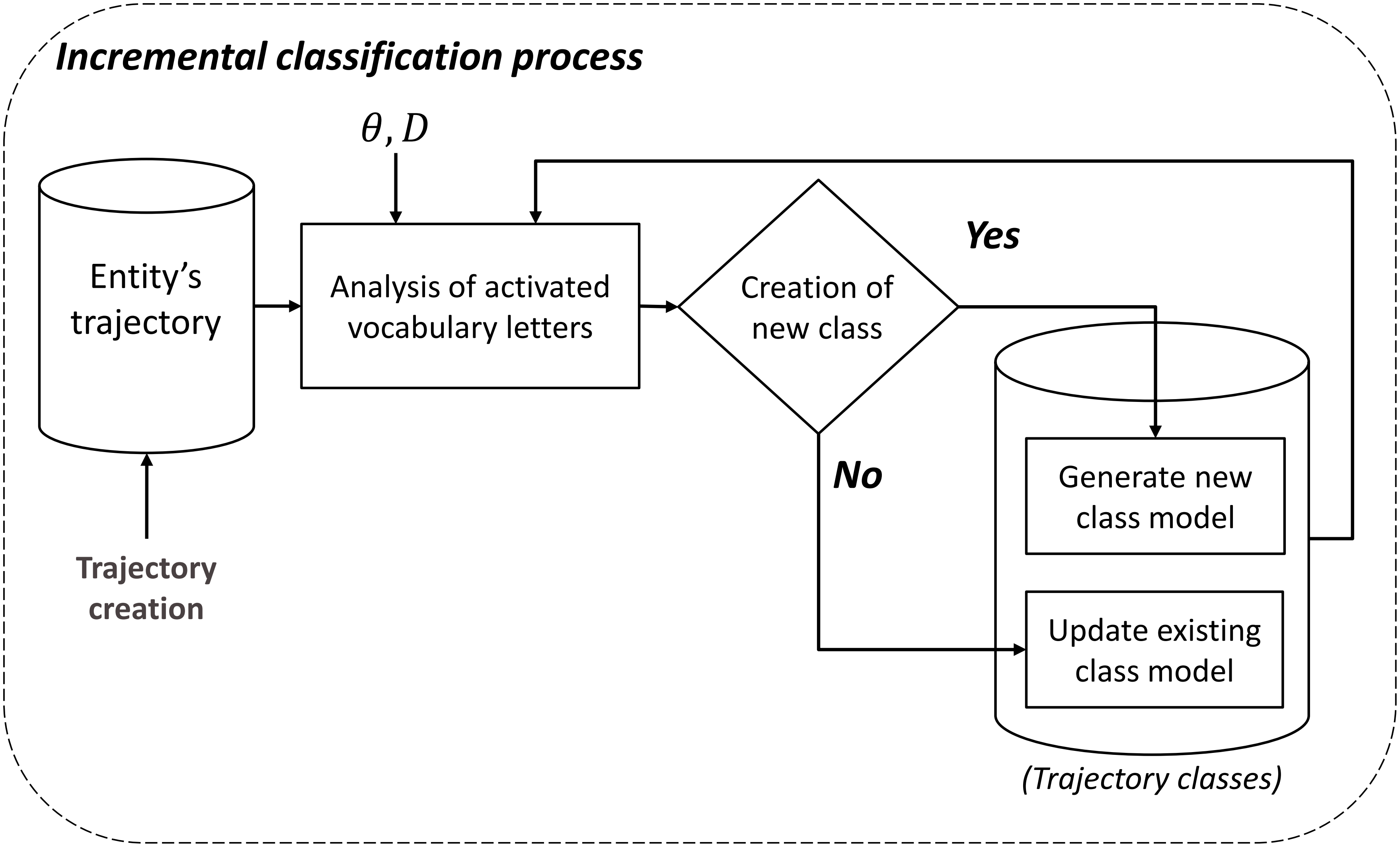


Summarizing (vocabulary creation)

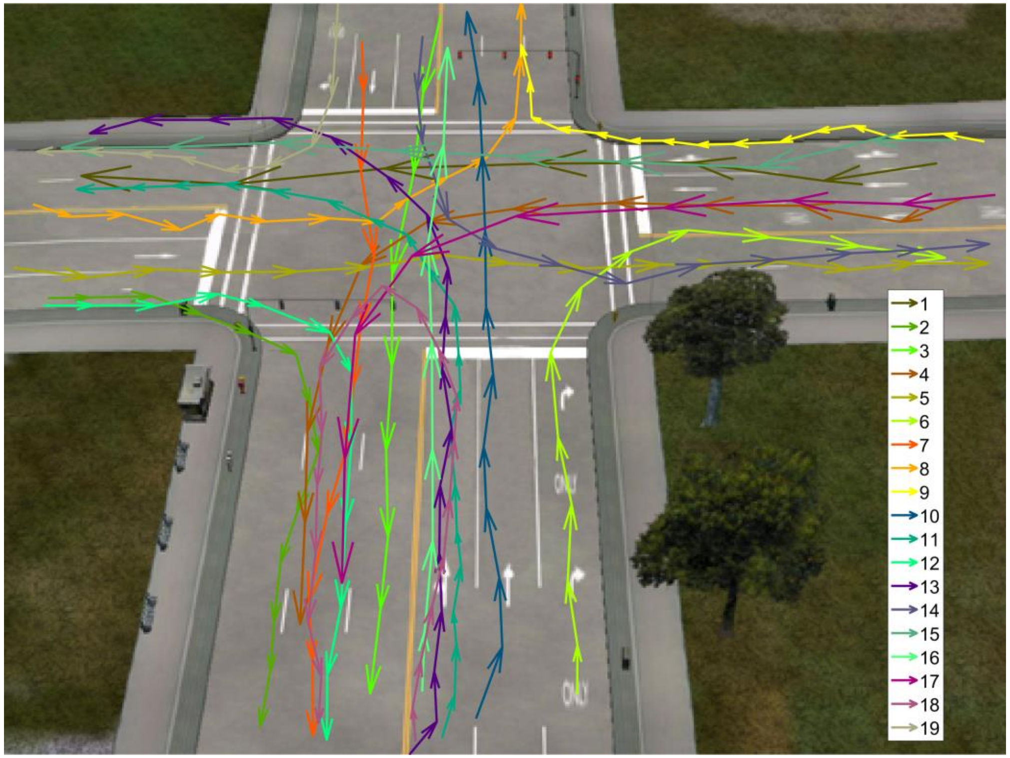
Offline vocabulary learning



Summarizing (dictionary creation)



Simulated data



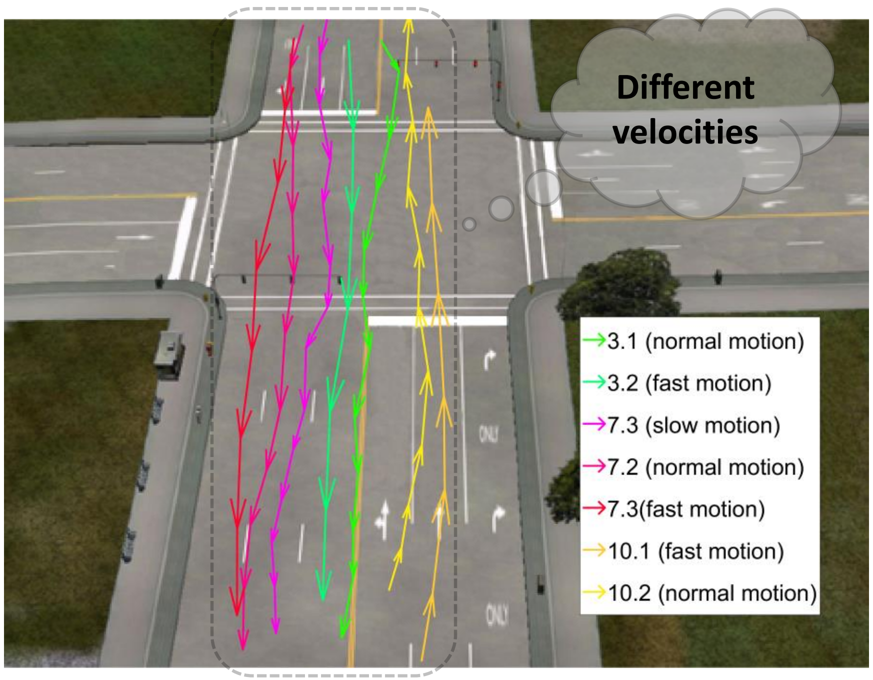
The CROSS dataset [1] is a simulated environment where objects move according to **19 classes** (words) proposed by authors.

- Each class contains **100 tracks** designed for **training** models and **500 trajectories** for **testing** them.
- **Training tracks** are used to build the **vocabulary**.
- **Testing trajectories** are used to generate the **dictionary** (**classes**).

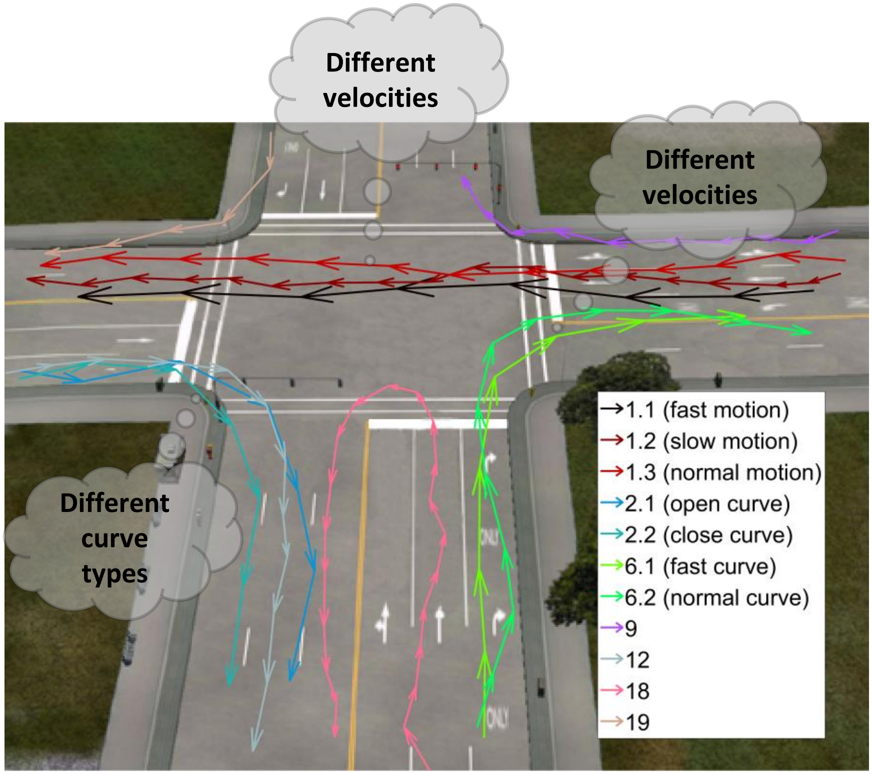
[1] B. Morris and M. Trivedi, "Learning trajectory patterns by clustering: Experimental studies and comparative evaluation," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2009, pp. 312–319.

Results

The proposed algorithm **found 47 trajectory classes (words)** in an unsupervised way. Such number differs from the **19 proposed classes** due to the **inclusion of velocity** and **time spent** in the video sequences.



Subclasses generated for three ground truth classes



Subclasses generated for six ground truth classes

Confusion matrix for simulated data

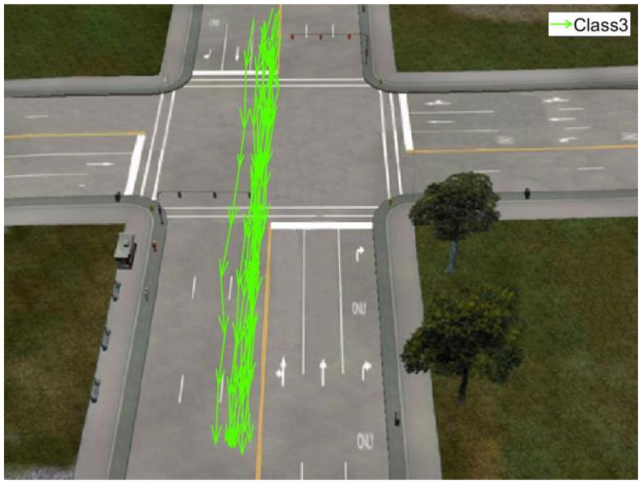
Classes

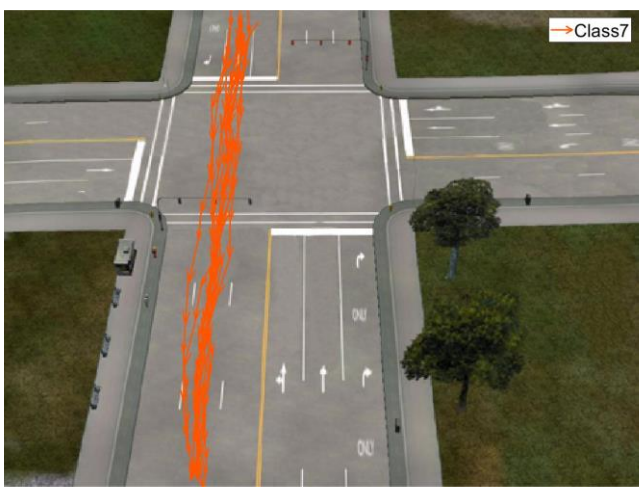
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	72.6	0	0	0	0	0	0	0	0.8	0	0	0	0	0	26.6	0	0	0	0
2	0	98.2	0	0	0	0	0	0	0	0	0	1.8	0	0	0	0	0	0	0
3	0	0	79.6	0	0	0	20.4	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	99	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	38.4	0	0	0	61.6	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	98	0	0	0	0	0	2	0	0	0
11	0	0	0	0	0	0	0	0	0	0	40	0	60	0	0	0	0	0	0
12	0	10	0	0	0	0	0	0	0	0	0	90	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	34	0	62.8	0	0	3.2	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
15	14.8	0	0	0	0	0	0	0	7.8	0	0	0	0	0	77.4	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
17	0	0	0	20.2	0	0	0	0	0	0	0	0	0	0	0	0	79.8	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Maximum confusion is obtained between couples of classes **3-7** and **11-13**

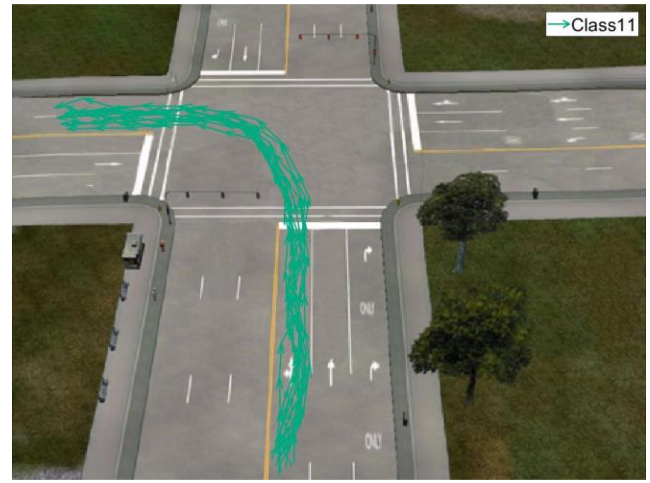
Maximum confusion cases

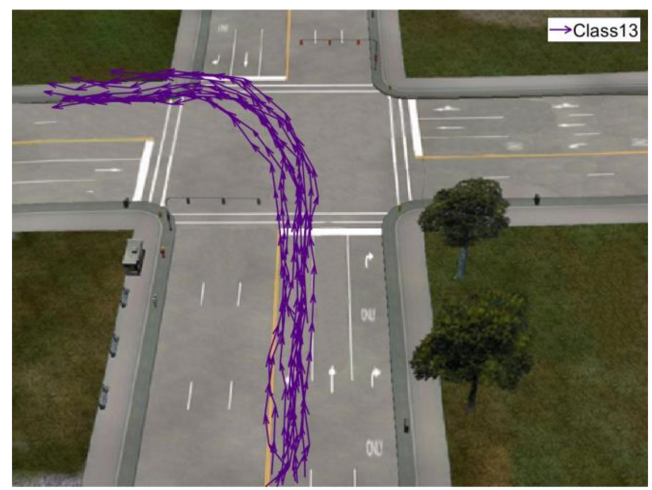
Random testing trajectories
classes **7-3**





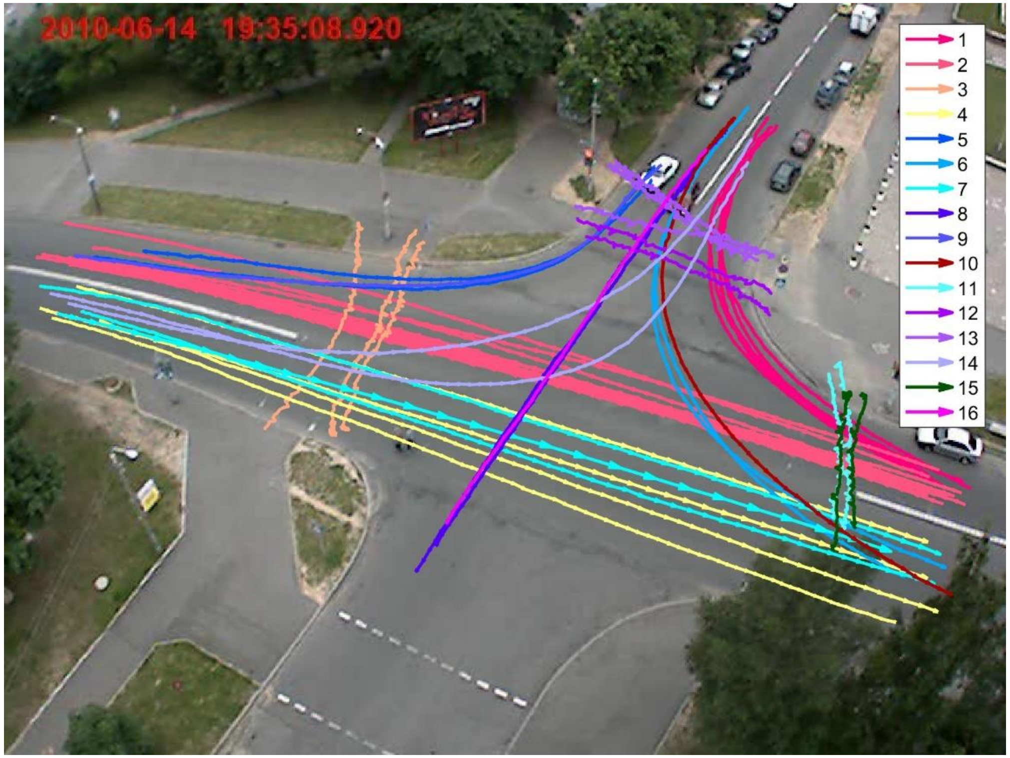
Random testing trajectories
classes **11-13**





[1] B. Morris and M. Trivedi, “Learning trajectory patterns by clustering: Experimental studies and comparative evaluation,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2009, pp. 312–319.

Real data



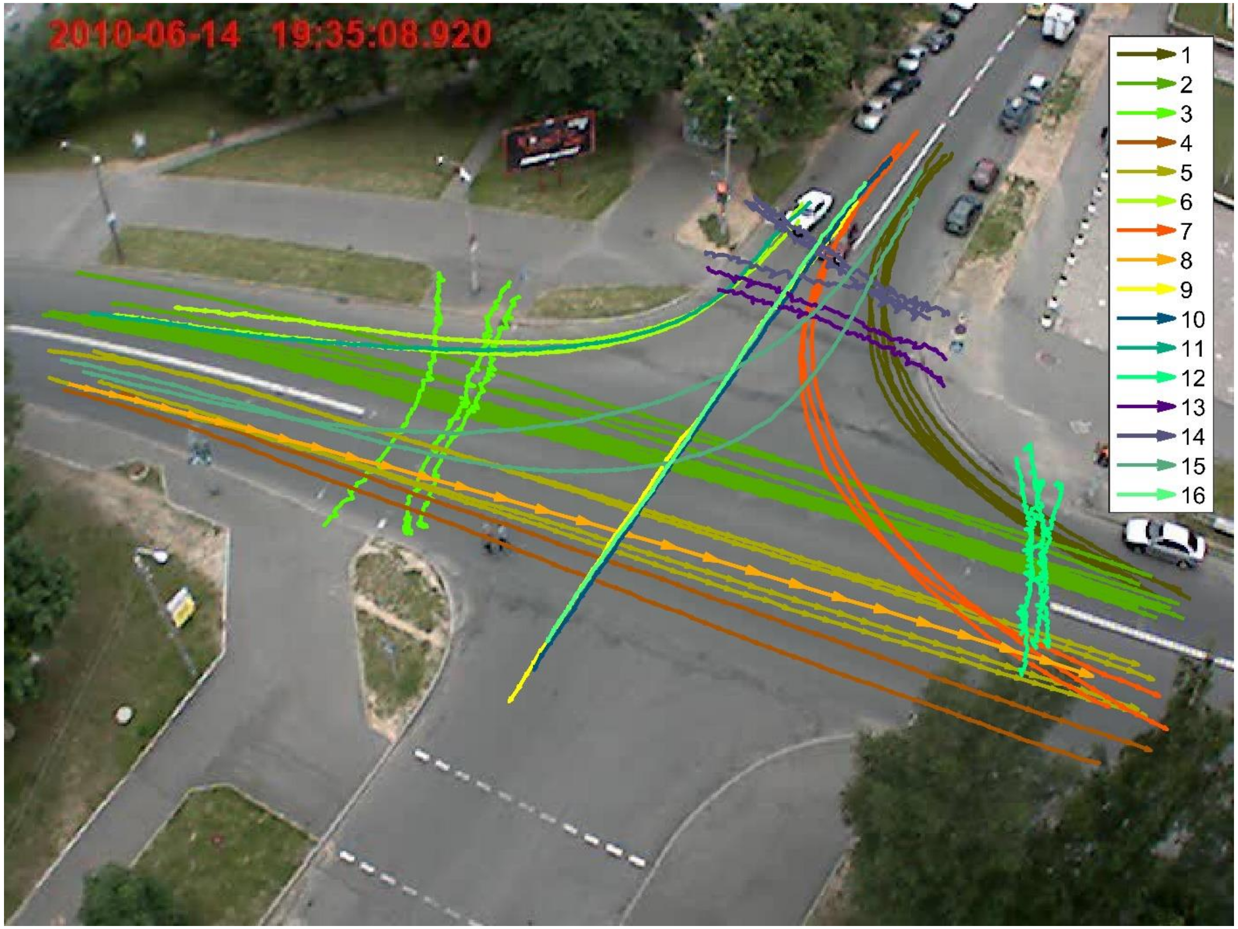
The PDTV dataset [2] consists of video sequences contains **51 trajectories** organized in **16 classes** (words) proposed by authors.

All 51 trajectories are used to build the **vocabulary** and the **dictionary.**

[2] N. Saunier et al, “A public video dataset for road transportation applications,” in TRB 2014 Annual Meeting, Washington, D.C., January 2014, p. 17.

Results

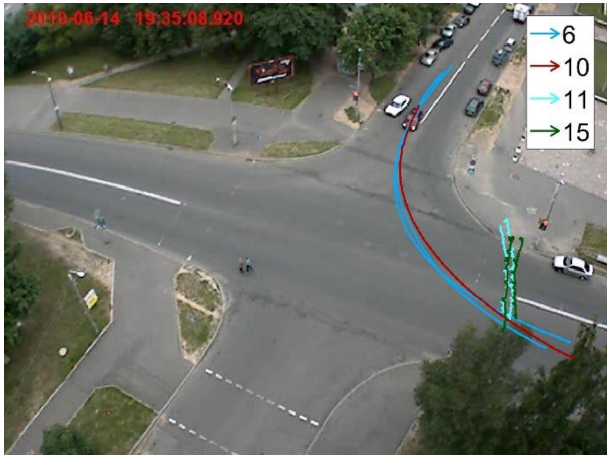
As proposed by the ground truth, **16 classes** (words) were also obtained by our algorithm. Nonetheless, the **acquired labels are slightly different from the ground truth**.



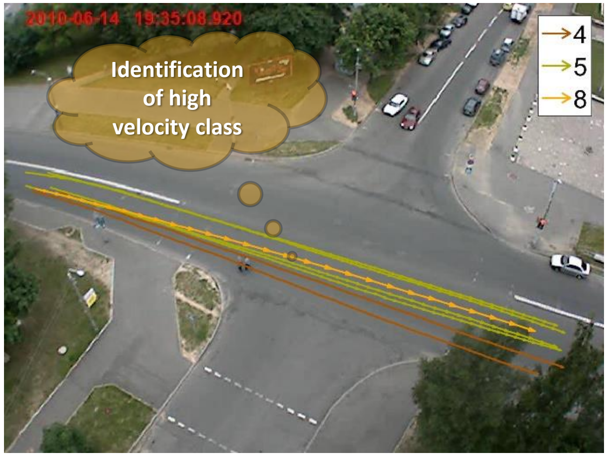
Comparison with proposed labels

Ground truth labels

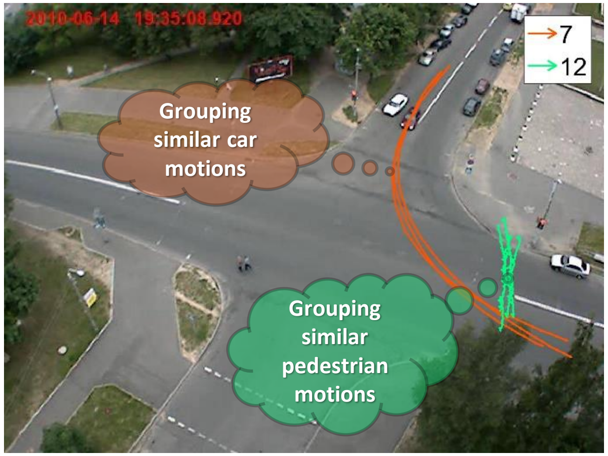
Ground truth labels



Obtained labels



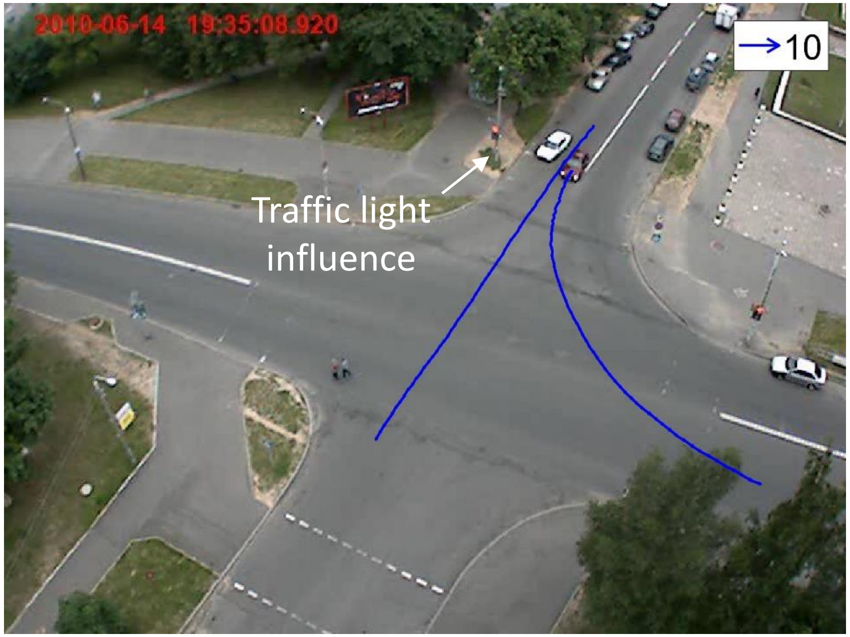
Obtained labels



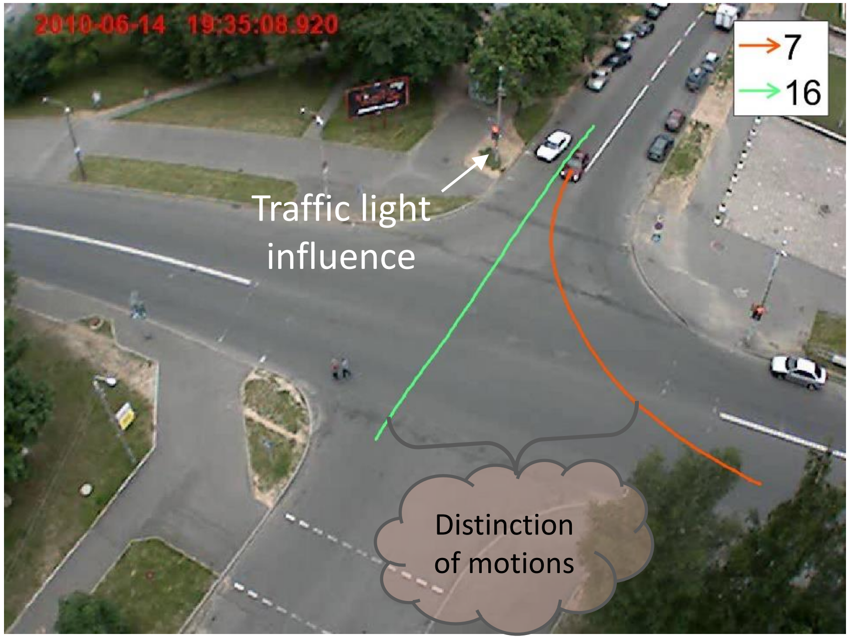
Advantages of proposed method

Vehicles stopped by the traffic light for long periods of time represent a problem for other classification algorithms [3] since they share **many similar points**.

Problematic classes in [3]



Classification of problematic classes in [3] (our method)



[3] V. Bastani, L. Marcenaro, and C. S. Regazzoni, “Online nonparametric bayesian activity mining and analysis from surveillance video,” IEEE Transactions on Image Processing, vol. 25, no. 5, pp. 2089–2102, May 2016.

Conclusions and future work

Our method for unsupervised trajectory clustering that uses a **weighted SOM** to generate a common **vocabulary** that encodes relevant trajectory information.

A distance matrix from the produced vocabulary to facilitate the incremental recognition of trajectory patterns (**words**) that can be used for **classifying unobserved trajectory data**. Results obtained with real and simulated data suggest that our method can generate detailed trajectory classes automatically.

Our approach enables the obtainment of a **dictionary of trajectories** based on their **location**, **velocities** and **time spent** a video sequence.

As a future work, we will employ **probabilistic filtering** that uses continuous and discrete information for tracking of objects in video data.

Thank you for your attention