Traffic Pattern Modeling, Trajectory Classification and Vehicle Tracking within Urban Intersections

Cheng-En Wu   Wen-Yen Yang   Hai-Che Ting   Jia-Shung Wang
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan
seraphim415@hotmail.com   wanyaneyang@yahoo.com.tw   ding3639849@gmail.com   jswang@cs.nthu.edu.tw

***Abstract*—Traffic behavioral monitoring within urban intersections is an essential issue in the Intelligent Transportation Systems (ITS) for a smart city. This paper investigates on gathering traffic information within an urban intersection where accidents frequently occur. In this paper, traffic pattern modeling, trajectory classification and a real-time vehicle tracker within the urban intersection are proposed. Trajectories of vehicles within the intersection are more regular than that of pedestrians; such monotonous trajectories can be classified with Hidden Markov Model (HMM) into various kinds of motion patterns and tracklet prediction can be performed then. So, given an identified prefix trajectory (for a new coming vehicle), the most likely model is determined and the probable template (tracklet) with the highest similarity is selected. This template gives the direction to forecast the next few locations the vehicle may pass through. Besides, tracking all of trajectories in real-time is a computational challenge, on the basis of vehicle movement and tracklet prediction, the proposed method can remove most of the unnecessary particles. The experimental results demonstrate both the computational effectiveness and tracking correctness of the proposed method, and the tracker truly executes in real-time for the intersections of six traffic lanes, say around six vehicles per second on tracking.**

***Index Terms*—Video surveillance, Vehicle tracking, Particle filter, Hidden Markov Model (HMM), Trajectory classification, Tracklet prediction.**

1 INTRODUCTION

Traffic behavioral monitoring is an important topic in Intelligent Transportation Systems (ITS) for a smart city. Useful information such as the percentage of a certain type of motion patterns, the average time to cross the intersection, etc. is helpful for organizing the structure of urban intersections that may reduce car accidents and avoid traffic congestions as well. As the number of cameras installed at urban intersections increases drastically, a lot of traffic surveillance videos will be generated endlessly. Such large amount of videos makes it impossible for a human to process properly. It is essential to design a traffic flow analysis system, which is able to collect traffic information automatically.

To collect the traffic flow information truthfully, it is necessary to extract every vehicle trajectory instantaneously. Thousands of vehicles pass through an urban intersection within an hour, most of them can be classified into limited types of motion patterns. Thus, vehicle trajectory classification plays an important role in traffic analysis.

The objective of this study is to build a traffic flow analysis system, including traffic pattern modeling, trajectory classification and a real-time multi-vehicle tracker within an urban intersection. Since our scene is fixed at urban intersections, some characteristics of trajectories in this area can be used. Because of the constraints of the road structure and traffic rule, vehicles tend to cross the intersection in a normal way. Thus, we may be able to predict the next moving locations of the vehicles. With the help of predicted locations, trackers can reduce the computational cost and handle traditional tracking problems like occlusion and illumination change.

We proposed approaches for trajectory classification and vehicle location prediction. Hidden Markov Model (HMM) for each type of motion patterns are trained and used for classifying new coming trajectories. While performing trajectory classification, a *k*-means like classification algorithm for better performance is proposed. In addition, we predict the vehicle location by classifying the prefix trajectory and find a most similar trajectory in templates set to complete the whole trajectory. The real-time multi-vehicle tracker can handle multiple objects of various shapes and sizes. Actually, this tracker can handle six vehicles in one second, in terms of computation time for one vehicle around 1/180 seconds. Since we want to collect information for road safety at intersections, we focus on outdoor urban traffic scenes. A typical scene with tracking results are shown in Fig. 1.

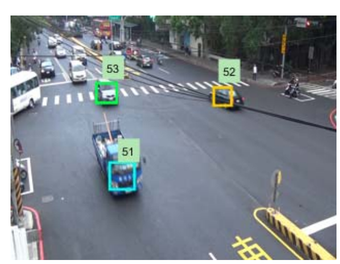

Fig. 1. Demonstration of tracking effect on a typical scene.

978-1-5386-2524-8/17/$31.00 ©2017 IEEE

2 Related Works

A. Tracking Methods

Firstly, there are many tracking algorithms for visual tracking. Recently, the popular approach is tracking-by-detection [1], which treats the tracking problem as a detection task over time. Sequentially, algorithms for adaptive tracking [2], [3] are proposed to make trackers more robust to pose variation and background clutter. Furthermore, Mei and Ling [[4] proposed a tracking approach based on sparse representation to handle the corrupted appearance and recently it has been further improved [5]. Moreover, in order to address occlusion, [8] employs a template update strategy, which combines incremental subspace learning and sparse representation.

These approaches help locate the target more accurately and are less insensitive to occlusion. However, all of the above are not flexible to incorporate in the external mechanism. Besides, considering the vehicle tracking within an intersection is relatively simple (background) and vehicles without seriously deformation, so we do not choose these complicated trackers. Moreover, real-time vehicle tracking is our essential goal, thus we are most concerned about computing performance. Therefore our tracker is particle-filter based.

Particle Filter

Particle filter is a hypothesis tracker [9], which is famous for simplicity, flexibility, and unsophisticated to implementation. According to Bayesian approach, the tracker relies on a sample-based representation of posterior probability density function (PDF). Weighted multiple particles (samples) are represented to the each state in terms of the likelihood function. An approximation of the variable of interest is obtained by the weighted sum of particles. The tracker has two major steps: prediction and update. During the prediction, each particle is evaluated based on given all available observations within the region of interest. In the update stage, particle filter uses probabilistic system transition model to predict the posterior, and then the weight of particles is re-evaluated with the new data.

B. Hidden Markov Models

Hidden Markov Model (HMM) is based on supervised learning and statistical modelling for temporal data [10]. This model has been used successfully in voice recognition, handwriting recognition and visual recognition of sign language [11]. Configuration of an HMM is shown in **Fig.** 2. The sample model is regarded as a graph with four internal and two marginal states connected by (oriented) transitions [12]. Our classification algorithm inspired by [13]. Perrone and Connel applied a *k*-means like HMM training method for allograph identification. The method refines the HMM parameters by iteratively reassign the allograph until no allograph labels have changed.

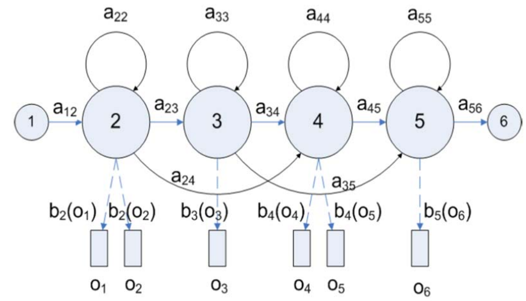

Fig. 2. Configuration of a Hidden Markov Model.

C. Trajectory classification

Xin, Yang, Chen and Li [14] claim that it is possible to pre-define all vehicle activities at a specific intersection because the signal control strategy and road structure are clearly known. After the number of vehicle activity is determined, a multi-layer spectral clustering approach is applied. Morris and Trivedi [15] developed a system for highway monitoring and activity analysis. The system learns the normal activities happened in live video and builds HMMs by the spatial and temporal characteristic of the vehicle trajectories. With the help of these models, abnormal trajectories can be detected and future intention of prefix trajectory can be predicted. Cai, Wang, Chen, and Jiang [16] also performed trajectory classification based on HMMs. In trajectory clustering phase, a coarse to the fine method is applied to learn motion patterns at intersections. The *k*-nearest neighbor algorithm is applied to classify prefix trajectory to the most similar motion pattern.

3 The Proposed Method

The proposed traffic flow analysis system within an urban intersection combines traffic pattern modeling, trajectory classification and a real-time vehicle tracker. As mentioned, trajectories of vehicles within the intersection are somewhat regular; such monotonous trajectories can be classified with HMM into various kinds of motion patterns thus tracklet prediction can be performed then. That is, given a just-recognized (by tracker) prefix trajectory, the most likely model is determined and the most probable template (tracklet) with the highest similarity is selected. This template gives the direction to forecast the next few locations. On the basis of vehicle movement and tracklet prediction, the proposed tracker removes most of the unnecessary particles. The complete flowchart of the proposed method is depicted in Fig. 3. The system flow can be described as the following eight steps:

Step 1. Initial clustering using sources and sinks.
Step 2. Motion pattern modeling by HMM (Baum-Welch algorithm) for each cluster.
Step 3. Model refinement: K-means clustering with HMM for all trajectories.
Step 4. Extract (using modified particle filter) the prefix trajectory for a new coming vehicle.
Step 5. Feed into every HMM models and query which motion pattern (model) it belongs to.
Step 6. Predict the next several moving locations with the corresponding trajectory we have learned.
Step 7. Send this location to the tracker.
Step 8. Do Steps 5-7 till the vehicle disappeared.

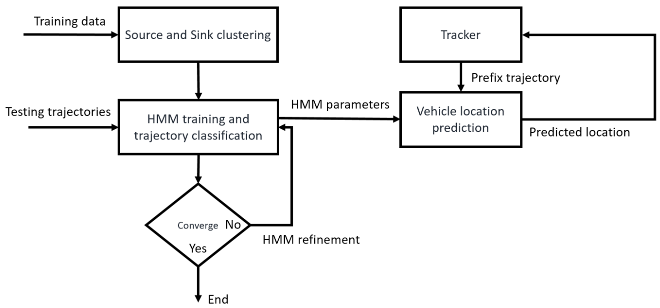

Fig. 3. The complete flowchart of the proposed method.

A. HMM Training and Trajectory Classification

As shown in the left part of Fig. 3, it consists of four procedures: source and sink clustering, HMM training, trajectory classification, and HMM refinement.

Initial clustering

Vehicle trajectory is constrained by the structure of urban intersection. We can coarsely classify the trajectories by their beginning (source) and ending (sink) points. Here, we apply *k*-means for clustering. The parameter *k* is determined by the structure of the urban intersection. Trajectories with the same source and sink are classified into the same motion pattern. In our experiment, eight kinds of motion patterns have been extracted in the training video sequence. Then, trajectories with the same motion pattern are used to train a HMM model. The diagram of eight motion patterns is depicted in Fig. 4.

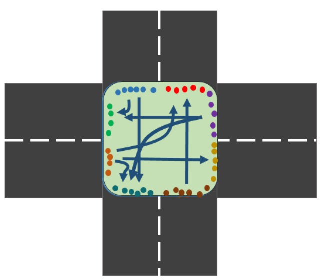

Fig. 4 Samples of eight motion patterns.

HMM model training

Each motion pattern is modeling by an HMM using the Baum-Welch algorithm. This model can be used to verify whether a trajectory follows the behavior of the motion pattern.

Trajectory classification and HMM refinement

After training, the HMMs can be applied to classify the new coming trajectory of a coming vehicle. A *simple* classification rule is to feed a trajectory into every HMM and decide which model it belongs to using the probabilities calculated by HMMs only. That is, it will be assigned to the HMM model (corresponding to a specific motion pattern) with the largest probability. In our experience, this *simple* rule could not perform well, so two additional steps are delivered to enhance the accuracy of classification. First, the *k*-means clustering with HMM iteratively refine the classification result till converge. Second, the canonical vector considers the similarity of observation in each state, see below.

This *simple* classification rule is described as: Given a new trajectory T of size 10 (Due to the simplicity of intersections, we limit the number of states to be 10), let the observed sequence be $O = [o_1, o_2, \ldots, o_{10}]$. Every observation is the moving direction between states. This *simple* rule calculates the probability of trajectory T under each HMM λ^k by:

$$\alpha_1^k(\mathrm{i}) = \pi_{\mathrm{j}}^k b_{\mathrm{j}}^k(o_1)(1 \le \mathrm{i} \le \mathrm{N} = 10)$$

$$\alpha_{t+1}^k(j) = \left[\sum_{\mathrm{i}=1}^{\mathrm{N}} \alpha_{\mathrm{t}}^k(i) a_{ij}^k\right] b_j^k(o_{t+1})$$

$$P(T|\lambda^k) = \sum_{i=1}^{N} \alpha_{\mathrm{t}}^k(i)$$

The HMM fits this trajectory best can be found by

$$\lambda^* = \arg\max_k P(T|\lambda^k)$$

The canonical vector is a supplementary step. It is activated when the probabilities calculated by HMMs are too close, thus the *simple* rule may trap into a wrong decision. Canonical vector is the most likely observations in the 10 states. When canonical vector is employed, the distance between canonical vector and trajectory observation is calculated. The trajectory is classified to the motion pattern whose canonical vector has the smallest distance with the trajectory observation.

K-means clustering with HMM is an iterative approach. First, the trajectories are classified by HMMs. Each trajectory is labeled according to their motion patterns. After that, the HMMs are retrained on the basis of the new label. The classification is terminated until no labels have changed. By this method, some wrong classified trajectories can be relabeled and assigned to the appropriate motion pattern.

Once a trajectory with all low probabilities for all models, It will be recognized as an *abnormal* trajectory.

B. Vehicle Location Prediction

Given a prefix trajectory (corresponding to a new coming vehicle). The flowchart of vehicle location prediction is shown in the right part of Fig. 3. Two procedures are applied to classify this prefix trajectory into a motion pattern first and then predict its future positions. After the tracker received the predict locations, actual location is calculated and returns to vehicle location prediction procedure. This mechanism is repeated iteratively until the vehicle leaves this intersection.

Prefix trajectory classification

The basic concept of prefix trajectory classification is the same as (complete) trajectory classification. The only variation is that we do not know how far the vehicle has moved, which make it hard to determine how many states should be considered. Thus, an extra procedure is considered to estimate how many states the vehicle has moved in this prefix trajectory. Accordingly, the HMM can be employed to classify the prefix trajectory into the proper motion pattern. The details are omitted here due to the limited space available.

Template selection and location prediction

After secure the motion pattern of the prefix trajectory as above, a template trajectory is located to predict the future locations, where the template trajectory is selected from the already-trained template set. This template set is a set of trajectories with various speed and moving directions obtained in training videos. We establish two constraints, speed constraint and moving direction constraint, to guarantee a proper template is selected from the template set. Also, the details are omitted here due to the limited space available.

C. The real-time tracker

As mentioned in Section 2, we choose the particle filter approach for solving the multi-vehicles tracking problem within urban intersections. To achieve real-time requirement, we first introduce two predictors to estimate the next possible location so as to eliminate large amount of particles. The first predictor is based on history information during vehicle tracking, and the second predictor is location prediction (HMM based) as discussed in Section 3.B. With these two predictors, we offer three procedures to deliberately reduce the number of particles and increase the tracking accuracy as well.

Predictor with sequential Bayesian filtering

Here, we present the first predictor with the help of sequential Bayesian filtering, where previous ten consecutive frames are employed to estimate next location.

Baseline search range of particles

To achieve real-time tracking requirement, we shrink the searching region through considering three kinds of cases. We have two predictors which can estimate two locations independently. As shown in left part of Fig. 5, there are the white point (predictor #2) and the blue point (predictor #1).

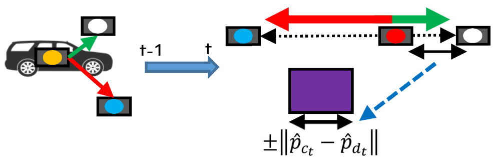

Fig. 5. Baseline range.

As Fig.5, the yellow point (t-1) with two distances between two points and then the new selected red point (t) is into equal distance of proportion as two arrows (t-1). Finally, the search region is relative to the distance of proportion. The following formulation details the search range (# of particles):

$$S_t = (2 \cdot \|\hat{p}_{c_t} - \hat{p}_{d_t}\| + 1)^2 \quad (1)$$

where $\hat{p}_{c_t}$ in Eq. 2 denotes a final selected prediction center of the search range:

$$\hat{p}_{c_t}: \frac{\|p_{o_{t-1}} - p_{d_{t-1}}\|_2 \cdot \hat{p}_{H_t} + \|p_{o_{t-1}} - p_{H_{t-1}}\|_2 \cdot \hat{p}_{d_t}}{\|p_{o_{t-1}} - p_{d_{t-1}}\|_2 + \|p_{o_{t-1}} - p_{H_{t-1}}\|_2} \quad (2)$$

in Eq. 2, where $p_{d_{t-1}}$ denotes mean movement of x-direction and y-direction, and $p_{o_{t-1}}$ denotes measurement position by the particle filter in previous frames. If search regions from two predictors are not overlapping, one of predictors would be inaccurate. Thus, there are two kinds of search regions in this case. The size of search range in terms of the number of particles as:

$$S_t = H_t + D_t \quad (3)$$

where H_t denotes the number of particles in the search range provided by location prediction (HMM):

$$H_t = (2 \cdot C_{HMM} + 1)^2 \quad (4)$$

and D_t in Eq. 5 denotes the number of particles in the search range provided by previous tracking vehicle movement mean variation:

$$D_t = (2 \cdot \|p_{o_{t-1}} - p_{d_{t-1}}\| + 1)^2 \quad (5)$$

In Eq. 4, C_{HMM} is the confidence interval (2σ) by location prediction (HMM). A measurement position by the particle filter in the previous frames $p_{o_{t-1}}$ and the prediction position by temporal derivations (mean acceleration) in previous frame $p_{d_{t-1}}$ are defined in Eq. 2.

Removing unlikely particles in a lane

Tracking in a video sequence, consecutive frames are quite similar, thus we can take this advantage of removing unlikely

Fig. 6. Demonstration of unlikely particles (purple pixels).

particles through overlapping between frames. In experiments, the result as illustrated in Fig. 6 shows that we truly find some unlikely particles (purple pixels) between two frames, during vehicle tracking.

Uncommon movement regulating

To increase the tracking accuracy, we adjust the location of the bounding box when an uncommon acceleration occurs. As illustrated in Fig. 7, which presents the red bounding box with the greater movement than the vehicle. To address this problem that has incorrect variation of movement of bounding box, we set a threshold to restrain the vehicle movement variation. In experiments, as shown in Fig. 8, when the bounding box occurs uncommon movement, which is greater than two standard deviation (2σ) of history record, we relocate this inappropriate bound box then, as shown in the lower level of Fig. 8.

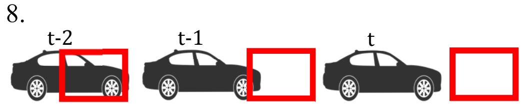

Fig. 7. Illustration of uncommon movement.

Fig. 8. Illustration of uncommon movement regulating in real cases.

5 Experimental Results

The proposed algorithms are implemented in MATLAB and running on a Core i5 3.20 GHz PC with 16GB RAM. In our experiments, we evaluate the performance of the proposed vehicle tracker, the accuracy of trajectory classification, and the precision of vehicle location prediction. The test scenes are from the intersection of Guangfu Rd. and Jiangong Rd., Hsinchu City. There are total 810 frames in this video sequence of size 320 x 240, in which total 65 vehicles (including trucks, sedans and motorcycles) passing through this intersection.

A. Performance of the Proposed Tracker

To evaluate the performance of the proposed tracker, we concentrate on three issues: accuracy, the execution time and the FPS. We compared the proposed algorithm (P-PF) with

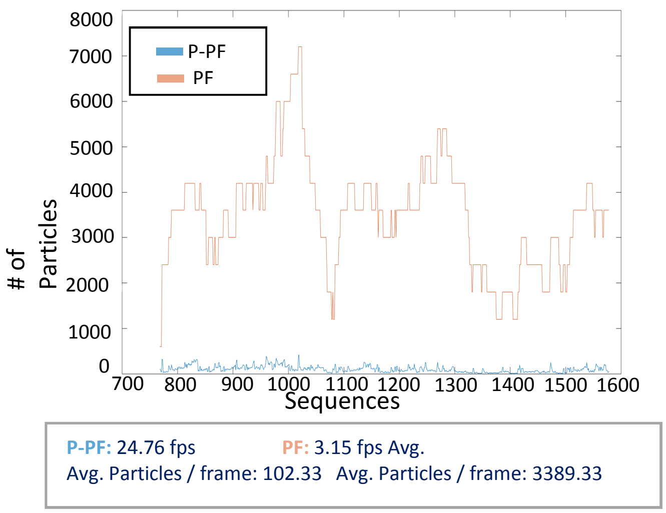

Fig. 9. Comparisons between P-PF and PF.

original particle filter (PF), as shown in Fig. . The speed of P-PF is greatly faster than PF, because the average quantities of particles by a frame is significantly lower than that of PF. To further boost the speedup of our tracker, we offer a skip mode, which sampling once every three frames, which indicates that the Skip-P-PF meets the requirement of real-time, 31.4 frames per second. Accuracy is measured by MOTA (Multiple Object Tracking Accuracy as follows:

$$1-\frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t},$$

where m_t, fp_t, mme_t, and g_t denote false negative (miss), false positive, mismatch and ground truth at frame t. As shown in Table 1, our accuracy of both proposed tracker achieves over 90%. Some fail tracking results are shown in Fig. 10.

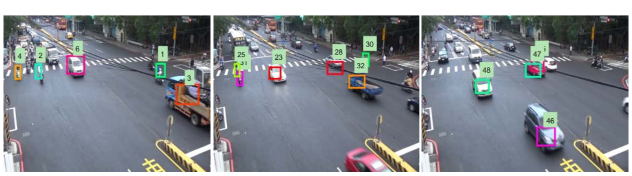

Fig. 10. Some failed tracking cases.

where left plot represents the failed case due to background clutter; middle plot shows that the movement of the prediction is slower than the vehicle; as for right plot, the bounding box drifts away when the target undergoes long-time occlusion.

Test Sequences	Tracker	MOTA (%)	Exec. time(s)	FPS(frame s/second)
East-West direction (810 frames)	P-PF	93.3	32.7	24.8
	Skip-P-PF	92.8	25.8	31.4
	PF	93.8	256.9	3.2

Table 1. Comparisons between trackers.

B. Trajectory Classification

There are 230 trajectories are extracted from the video sequence. Among them, 115 trajectories are used for training and the other 115 trajectories are used for testing. One abnormal trajectory is erroneously classified to motion pattern 8, because they have the same beginning moving direction, which makes the HMMs confuse. The details of quantitative evaluation are shown in Table 2 and Table 3. Compared with the result of *simple* classification rule in Table 2, our method has a better performance. The proposed method (k-means clustering with HMM and canonical vector) enhances the recall rate and precision rate in Motion pattern2, Motion pattern3 and Motion pattern 5. By applying our proposed method, the total classification rate (recall) increases from 90.34% to 97.39%.

Pattern	1	2	3	4	5	6	7	8	*
Recall	100	100	100	100	71	100	100	100	67
Precision	100	100	100	100	100	100	100	50	50
F1	100	100	100	100	83	100	100	67	57

Table 2. Classification rates with the *simple* classification rule. (*abnormal)

Pattern	1	2	3	4	5	6	7	8	*
Recall	100	80	86	100	57	100	100	100	67
Precision	100	91	100	100	100	100	100	50	20
F1	100	85	92	100	73	100	100	67	31

Table 3. Classification rates with our proposed method.

C. Vehicle Location Prediction

The performance of prefix trajectory classification is discussed and some results of the predicted vehicle locations are shown. We evaluate prefix trajectory classification in three kinds of ratio, 0.2, 0.5, and 0.8. For example, when the ratio is 0.2, it means that the first 0.2 portion of trajectory is set as prefix trajectory. Classification accuracy is low when a small ratio of trajectory is given. Consider Motion pattern 2 and Motion pattern 3 as examples, these two motion patterns have the same moving direction in the beginning, it is hard to distinguish them while only a small portion of trajectory is given. As the given trajectory becomes longer, the classification rate increases as well. The summary of prefix trajectory classification is shown in Table 4. The result shows that when half of the trajectory is given, we have pretty high confidence to classify it to the right pattern.

Prefix ratio	0.2	0.5	0.8
Video sequence	0.66	0.8	0.92

Table 4. Prefix trajectory classification rate (Recall).

Some typical cases of future location prediction are demonstrated in Fig. 11. The red points denote the given prefix trajectories, green points denote the predicted results, and the purple points are the ground truth. ‘

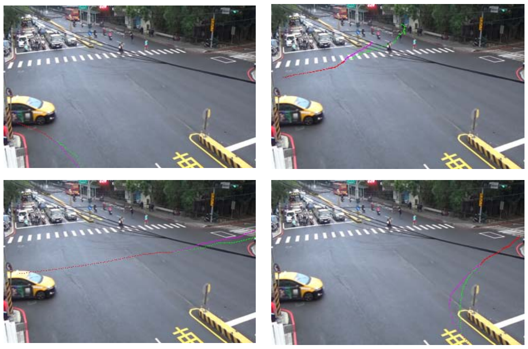

Fig. 11.Prediction of vehicle locations (ratio = 0.5).

To evaluate the effectiveness of our algorithm, there is an experiment in second intersection. The result is on the website. https://sites.google.com/site/trajectoryclassificationeg2/

6 Conclusion and Future Work

This paper presents a traffic flow analysis system, including traffic pattern modeling, trajectory classification, vehicle location prediction and a real-time multi-vehicle tracker for video surveillance at urban intersections. First, a procedure for classifying the trajectories into corresponding motion pattern is proposed. We utilize *k*-means clustering with HMM to classify the trajectories iteratively, which is able to refine the wrong assignments. A matching with canonical vector is activated while the HMMs are confused. And, a procedure for forecasting the future locations of vehicles is presented. By predicting the states of the prefix trajectory, HMMs can be employed to classify prefix trajectories. Then, a proper template is selected and the future locations of vehicles can be projected accordingly.

The experimental results show that the performance of trajectory classification is upgraded 7% compared with the *simple* classification rule. The vehicle location prediction performs well in most of the cases and can be helpful for vehicle tracking.

The real-time tracker is based on the particle filter and coupled with HMM-based vehicle location prediction. To improve efficiency and accuracy, we introduce three procedures to remove unnecessary particles and adjust error from two predictors. Experimental results show that the tracker truly executes in real-time for the intersections of six traffic lanes, say around six vehicles per second.

In the urban city, our system can provide a real-time warning to neighboring vehicles when abnormal trajectories of vehicles occur at intersections. In the future work, the deep learning method will be employed to train our proposed tracker to further improve prediction accuracy, also we will implement our algorithm in C, which can speedup the computing performance to handle more vehicles at intersections in real-time.

7 References

[1] S. Avidan. “Support vector tracking.”*IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, Issue: 8, August 2004, pp.1064-1072.

[2] A. Saffari, M. Godec, T. Pock, C. Leistner and H. Bischof “Online multi-class LPboost. ”*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp.3570-3577.

[3] H. Grabner, C. Leistner, and H. Bischof, “Semi-supervised on-line boosting for robust tracking. ”*European conference on computer vision (ECCV)*, October 2008, pp.234-247.

[4] X. Mei, and H. Ling “Robust visual tracking using L1 minimization”. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, September 2009, pp.1436-1443.

[5] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, “Minimum Error Bounded Efficient L1 Tracker with Occlusion Detection”. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011, pp. 1257-1264.

[6] R. Sen, A. Maurya, B Raman, et al, “Kyun queue: a sensor network system to monitor road traffic queues”. *ACM Conference on Embedded Network Sensor Systems*, November 2012, pp. 127-140.

[7] X. Jia, H. Lu, M.H. Yang, “Vehicle Tracking Using Projective Particle Filter.” *IEEE Conference on Advanced Video and Signal Based Surveillance*, October 2009, pp. 7-12.

[8] X. Jia, H. Lu, M.H. Yang, “Visual tracking via adaptive structural local sparse appearance model.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 1822-1829.

[9] K.V. Tangirala, and K.R. Namuduri, “Object Tracking in Video Using Particle Filtering.” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP),* March 2005, Vol. 2, pp.657-660.

[10] C.M. Bishop, “Sequential Data,” *in Pattern Recognition and Machine Learning*, New York: Springer, 2006, pp.610-614

[11] T.E. Starner. “Visual Recognition of American Sign Language Using Hidden Markov Models,” Brain and Cognitive Sciences dept., Massachusetts Institute of Technology, Cambridge, MA, Rep., 1995.

[12] L. Rabiner, and B. Juang “An introduction to hidden Markov models. ” *IEEE ASSP Magazine,* Janunary 1986, Volume:3, Issue: 1, pp. 4-16.

[13] M.P. Perrone, and S.D. Connell, “K-means clustering for hidden Markov models.” *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, 2000, pp. 229-238.

[14] L. Xin, D. Yang, Y. Chen, and Z. Li, “Traffic flow characteristic analysis at intersections from multi-layer spectral clustering of motion patterns using raw vehicle trajectory.” *International IEEE Conference on Intelligent Transportation Systems*, 2011, pp. 513-519.

[15] B.T. Morris, and M.M. Trivedi, “Learning, modeling, and classification of vehicle track patterns from live video.” *IEEE Transactions on Intelligent Transportation Systems*, 2008, vol. 9, pp 425-437.

[16] Y. Cai, H. Wang, X. Chen, and H. Jiang, “Trajectory-based anomalous behaviour detection for intelligent traffic surveillance.” *IET Intelligent Transport Systems*, 2015, vol 9, pp. 810-816.