

# Probabilistic Modeling of Dynamic Traffic Flow between Non-overlapping FOVs

<sup>†</sup>Wei-Chen Chiu (邱維辰), <sup>‡</sup>Ching-Chun Huang (黃敬群),

<sup>†</sup>Jen-Hui Chuang (莊仁輝), <sup>‡</sup>Sheng-Jyh Wang (王聖智)

<sup>†</sup>Dept. of Computer Science, National Chiao Tung University

<sup>‡</sup>Dept. of Electronics Engineering, National Chiao Tung University

E-mail: walon.cs97g@nctu.edu.tw

## ABSTRACT

The ability to infer the traffic status across multiple cameras allows the extended use of existing vision-based surveillance systems to global traffic monitoring. In this paper, we proposed an efficient algorithm to probabilistically model the dynamic traffic flow between non-overlapping FOVs. By assuming the transition time of object moving across cameras follows a global model and consecutively estimate the model parameters, we may infer the time-varying traffic status in the unseen region. In principle, the parameters of the transition time model can be estimated if the object correspondence between non-overlapping FOVs is known. However, object correspondence itself is still an unsolved problem in the literature. In this paper, we model object correspondence and the parameters estimation as a unified problem in a proposed Expectation-Maximization (EM) based framework. By treating object correspondence as a latent random variable, our proposed framework can iteratively search for the optimal object correspondence and model parameters. Experimental results on real data show the accuracy of dynamic model estimation and the beneficial inference of the traffic status.

**Keywords:** Traffic flow, Non-overlapping FOVs.

## 1. INTRODUCTION

The primary purpose of Intelligent Transport System (ITS) is to address the problems of road safety and congestion, which are important in modern traffic systems. ITS varies in a wide range of ap-

plications, from basic traffic management systems, such as car navigation and traffic signal control system, to advanced functions, like prediction of transit vehicle arrival time. However, no matter what kind of application it is, the key ingredient of ITS is the modeling of traffic flow to express the characteristics of traffic state. The basic idea of traffic flow modeling is to model the microscopic and macroscopic relationships among the traffic stream variables, including speed, flow and concentration. Many models were proposed as analytical techniques, such as traffic stream models including Greenshield's and Edie's model, shock wave analysis and queuing theories [1]. Consider an easier traffic flow modeling problem between two separated regions. Given a vehicle moving from one region toward another, its traveling time might increase due to a heavy traffic state associated with an accident or traffic jam, or might decrease in off-peak hours or due to other phenomena related to smooth traffic flow. Therefore, instead of using lots of variables to model the traffic flow, we can infer the traffic state by observing the "delay times" or "transition time" of objects moving across different regions.

Without loss of generality, the current methods for traffic flow monitoring can be roughly categorized into non-vision-based and vision-based approaches. For non-vision-based methods, most use an automatic vehicle location system (AVLS) to collect data, which include time and location pairs. However, an AVLS system has the drawbacks of being expensive in installation and maintenance. Besides, due to the privacy issue, prompting those AVLS-based applications is not as easy as it appears. Therefore, in this paper we will focus on

vision-based approach to deal with the problem of traffic flow monitoring. This is because the main type of equipment is the surveillance cameras with the features of relatively low installation cost, easy operation and little traffic disruption during maintenance in contrast to AVLS [2,3]. Moreover, many image processing and computer vision techniques for the analysis of traffic flow video sequences have been widely used in traffic flow monitoring. Nevertheless, almost all the existing vision-based traffic surveillance systems are limited by cameras' field of views (FOVs). Moreover, these systems only concern about the visible information in the video sequences and thus only indicate of the "local" traffic status at the camera location. For a camera surveillance system that covers a wide area, without the traffic status in the unseen region between non-overlapping cameras, we cannot infer a "global" traffic flow. Hence, if we want to build a practical vision-based wide-area surveillance system for traffic flow monitoring, we have to estimate the traffic state in the unseen regions between FOVs.

In this paper, we present an algorithm to infer the traffic state between non-overlapping FOVs by modeling the traffic flow with a transition time distribution, which may dynamically change. Our method provides a way to extend existing vision-based surveillance systems from local-range monitoring toward global-range monitoring of traffic flow. A simple interface of our system is shown in Figure 1. Moreover, if we can combine our method with the online map service, such as Google Map or the CCTV Live-cam service on some important traffic spots, then users can fetch the current traffic state from the Internet. This could be a potential application in the near future.

## 2. PREVIOUS WORK

In general, the key problem of surveillance systems with non-overlapping cameras is to build the relationships between objects moving through FOVs, i.e., the association of objects across cameras. Once the association is established for all objects, problems such as co-operative object tracking, object counting, and monitoring of objects activities become easier to resolve.

In [4, 5], Javed et al. proposed a method to establish the correspondence between observations across cameras. They learned the inter-camera il-

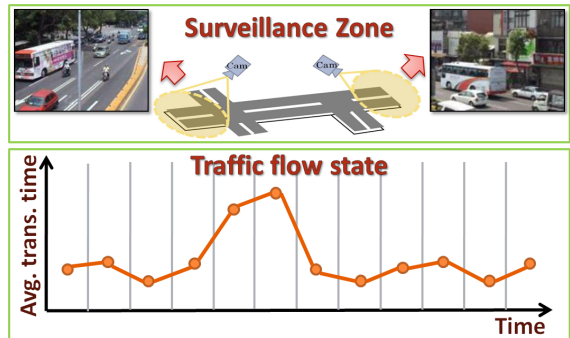


Figure 1: A simple interface of proposed system. It provides the information of the time-variant traffic state between non-overlapping FOVs.

lumination and transition properties via the appearance and space-time cues during a training phase. With these features, tracks of targets moving through FOVs are corresponded by maximizing the posterior probability. The method in [6] proposed by Song et al. used feature matching for tracking applications in a camera network. They treated similarities between features (appearance and biometric) as random variables, whose probability distribution were created via supervised learning. By building the feature graph containing the feature vector and similarity score observed over space and time, tracks of people can be found as the optimal paths in this graph.

Some works have been focused on recovering tracks and the poses between cameras with non-overlapping FOVs. Rahimi et al., in [7], reconstructed the trajectory of a target across non-overlapping cameras and simultaneously computed external calibration parameters of cameras, with the assumption that the target's dynamic state of location and velocity will evolve according to linear Gaussian Markov dynamics. In [8], Sheikh et al. presented a unified framework for the association of multiple objects across multiple cameras in planar scenes. The relationships between cameras are modeled by homographies and the object's trajectory are assumed to follow a polynomial kinetic model. They recovered the assignment of associations, while simultaneously computing the maximum likelihood estimates of the inter-camera homographies and the trajectory parameters by using the Expectation Maximization algorithm.

Instead of using the appearance feature or object motion model to directly infer the correspondence of objects across non-overlapping cameras, some research works try to recover the topology of a number of cameras based on co-occurrence of entries and exits. During the process of topology recovery, some indirect information about the correspondence of objects can also be extracted. In [9], Makris et al. assumed a single mode transition time distribution between FOVs and exhaustively search for the location of the mode by applying cross-correlation to the arrival and departure times of objects across cameras. This approach has been extended by Stauffer [10] and Tieu et al. [11] by providing a more rigorous definition of a transition based on information-theoretic mutual information, which can compute the statistical dependence between observations across cameras.

However, most of the features or models used in the above methods are insufficient in real life scenario for traffic flow surveillance. For example, in the wide-area surveillance, the appearance model may become invalid because the observation may occupy only a few pixels or looks different from different viewing angles of cameras. Likewise, for vehicles in the real life traffic which might contain complicated behavior patterns such as car-following and queuing, the aforementioned object motion model won't be able to provide accurate descriptions. Moreover, due to the dynamic changes of real life traffic, the transition time distribution may not be found by cross-correlation or by maximizing the dependence between observations in two cameras.

In this paper, we assume that all the vehicles in the unseen region between two FOVs will follow a global transition-time distribution, which can be used to describe the traffic flow. By observing the dependence between the correspondence of vehicles across cameras and the global transition time distribution, we present an algorithm to describe the traffic state adaptively over time.

### 3. PROPOSED METHOD

In this section, we first introduce the interesting targets we select for the analysis of traffic status in our system. By utilizing the specific motion pattern of the interesting targets, we can reduce the complexity of traffic flow analysis and improve

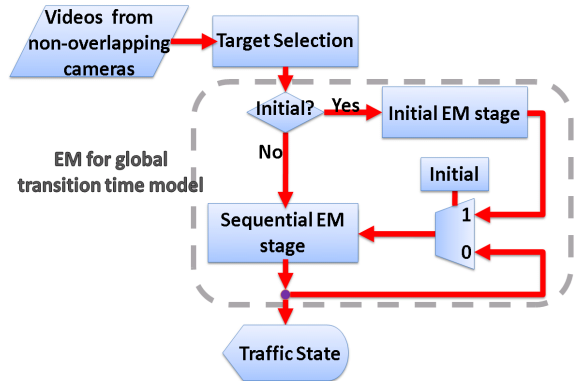


Figure 2: The block diagram of the proposed system.

the system performance. Then we will describe the traffic flow model of our system in Subsection 3.2. In Subsection 3.3, the problem formulation and our algorithm are presented. In Figure 2, we show the flow chart of the proposed system.

#### 3.1. Selected Target

Our goal is to monitor the traffic state between two non-overlapping FOVs linked by a road. In general, since there might be some intersections along the road, vehicles may leave the road in-between. To alleviate the complexity of analysis, we prefer to choose the vehicles with fixed route (e.g., bus) across FOVs as our main targets. Although we still cannot guarantee all the buses seen in one FOV will definitely show up in another, this target selection indeed reduces many burdens of our analysis. By selecting buses from the traffic videos taken from FOVs, we record the entry/exit times of all the buses in the regions of interest. A result of the target selection is shown in Figure 3.

#### 3.2. Traffic Flow Model

As we know, from a microscopic point of view, every vehicle has its own behavior pattern, which is affected by individual driver. However, if we observe the traffic flow from a macroscopic point of view, we will find that most vehicles follow a global trend due to the interactions between vehicles such as car-following and queuing. Hence, the main assumption of our proposed method is that the transition time of most vehicles across two



Figure 3: A result of target selection. We choose buses as our targets and record the exit time ( $\{x_m\}_{m=1\sim M}$ ) and entry time ( $\{y_n\}_{n=1\sim N}$ ) of them.

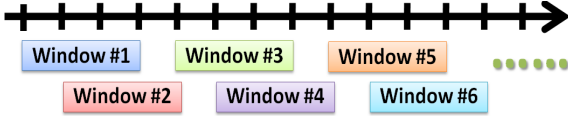


Figure 4: Divide the time-line into overlapped time-windows, and apply the traffic flow analysis at different times of the day.

non-overlapping FOVs will satisfy a global model. Here we use the Gaussian mixture model (GMM) to approximate this distribution because this model is general and it may approximate any continuous function under regular conditions. Then our objective is to derive the parameters of this global transition time model to describe the traffic state. Moreover, since traffic flow state may dynamically change over time, we divide the time-line into many overlapped time-windows to do the analysis of transition time distributions at different times of the day, as shown in Figure 4.

### 3.3. Problem Formulation

Before discussing how to determine the GMM parameters, we consider two opposite cases in the following paragraphs. First, suppose that we have the prior knowledge of the global transition-time distribution, we may infer the rough correspondence among buses. That is, we can approximately determine how the buses in two FOVs may match with one another. On the other hand, if we know how the buses in two FOVs match with one another and thus the transition time of each correspondence, the global transition time distribution can be derived. Hence, in our system, we exploit this

physical dependence between the correspondences of buses and the global transition-time distribution, and proposed a new solution to monitor the traffic between FOVs. Note that it is reasonable to suppose the traffic statuses on different sides of the road are independent of each other. Without loss of generality, we only discuss the traffic problems in a single side.

To determine the model parameters and to further infer the traffic state within a time window, we formulate the problem as an optimization process expressed as follows:

$$\Theta^* = \arg \max_{\Theta} p(\Theta|Z), \quad (1)$$

where  $Z = (\{x_m\}_M, \{y_n\}_N)$  represents the combination of observations, including the exit time set  $(x_1, x_2, \dots, x_m, \dots, x_M)$  and the entry time set  $(y_1, y_2, \dots, y_n, \dots, y_N)$  of all the buses in two non-overlapping FOVs. However, it is difficult to build the probability model  $P(\Theta|Z)$  in Equation 1 owing to the lack of physical connection between  $\Theta$  and  $Z$ . To compensate the physical gap between parameters  $\Theta$  and our observations  $Z$ , we introduce the correspondences between the exit time  $\{x_m\}_M$  and entry time  $\{y_n\}_N$  as an unknown random variable  $C = \{c_m(x_m) = y_n\}_M$ , where  $c_m(\cdot)$  indicates the entry time  $y_n$  that an exit time  $x_m$  will correspond to. Therefore, if a bus leaves one FOV at  $x_m$ , travels through the road, and then enters another FOV at  $y_n$ , we can express this correspondence by  $\{x_m, c_m(x_m) = y_n\}$ . The transition time of this correspondence is  $t_m = c_m(x_m) - x_m$ . By collecting all the transition times of each correspondence into a data set  $T = \{t_m\}_M$ , we may estimate the model parameters  $\Theta$ .

The problem, of course, is that we do not have the measurement of  $C$  and we cannot derive the  $\Theta$  directly. Hence, we treat the correspondence  $C$  as a latent variable and reformulate Equation 1 as Equation 2 by marginalizing out every possible correspondence  $C$ .

$$\Theta^* = \arg \max_{\Theta} \sum_C p(\Theta, C|Z). \quad (2)$$

Although a direct approach to computing this optimization problem is generally intractable, the Expectation Maximization algorithm provides a mean to do the maximization by iteratively calculating the following two steps:

**1. E Step:** Calculate the expected log likelihood function  $Q(\Theta)$ :

$$Q(\Theta) = \sum_C \log(p(Z, C|\Theta))p(C|Z, \Theta^{(old)}). \quad (3)$$

where the expectation is taken with respect to the posterior distribution  $p(C|Z, \Theta^{(old)})$  over all possible correspondences  $C$ , given the data  $Z$  and the  $\Theta^{(old)}$  at the previous time step.

**2. M step:** Find the Maximum Likelihood estimate  $\Theta^{(new)}$  by maximizing  $Q(\Theta)$ :

$$\Theta^{(new)} = \arg \max_{\Theta} Q(\Theta). \quad (4)$$

However in the E step, the expectation is difficult to compute since the number of possible correspondences explode combinatorially. A solution is to introduce Markov chain Monte Carlo (MCMC) to sample from the posterior probability distribution  $p(C|Z, \Theta^{(old)})$  and replace the expectation in the E step over all possible correspondences with MCMC samples. More Formally, this can be justified in the context of a Monte Carlo EM or MCEM. The sampling method we use here is the Gibbs sampling. In our setting, we prefer the entry/exit times to be one-to-one correspondence, but still allow the possibility that one entry time could be matched with more than one exit times. Therefore,  $p(c_j|c_1^{(r+1)}, \dots, c_{j-1}^{(r+1)}, c_{j+1}^{(r)}, \dots, c_m^{(r)}, \Theta)$ , the conditional probability of  $c_j$  given the other correspondences  $\{c_m\}_{M \neq j}$  and global transition time distribution  $\Theta$ , will be proportional to the likelihood of transition time  $t_j = c_j(x_j) - x_j$  determined by  $\Theta$ , while the conditional probability may decrease if the exit time  $y_n = c_j(x_j)$  has been matched with others (see Algorithm 1). By performing sampling  $R$  times, it will generate a sequence of  $C$ . Since we expect that the correct  $C$  will have a much higher likelihood than others, we will choose the best sampling result to enter the M step in EM algorithm.

With the correspondences  $C = \{c_m\}_M$  from the Gibbs sampling, we can calculate all the transition times  $\{t_m = c_m(x_m) - x_m\}_M$  of buses traveling across two non-overlapping FOVs. In the M step, from  $\{t_m\}_M$ , we want to derive the parameters  $\Theta$  of the global transition time distribution. Assume we use a GMM with  $K$  Gaussians to approximate the global transition time distribution, then the distribution  $\Theta$  will be parameterized on the weights  $w_j$ ,

---

**Algorithm 1** Gibbs sampler.

---

```

Initialize  $c_{(1:m)}^{(0)}$ 
for  $r = 0$  to  $R - 1$  do
  Sample  $c_1^{(r+1)} \sim p(c_1|c_2^{(r)}, c_3^{(r)}, \dots, c_m^{(r)}, \Theta)$ .
  Sample  $c_2^{(r+1)} \sim p(c_2|c_1^{(r+1)}, c_3^{(r)}, \dots, c_m^{(r)}, \Theta)$ .
   $\vdots$ 
  Sample  $c_j^{(r+1)} \sim$ 
     $p(c_j|c_1^{(r+1)}, \dots, c_{j-1}^{(r+1)}, c_{j+1}^{(r)}, \dots, c_m^{(r)}, \Theta)$ .
   $\vdots$ 
  Sample  $c_m^{(r+1)} \sim p(c_m|c_1^{(r+1)}, \dots, c_{m-1}^{(r+1)}, \Theta)$ .
end for

```

---

the mean  $\mu_j$ , and the variance  $\sigma_j^2$  for each Gaussian  $j$ , with  $j = 1, \dots, K$ . To estimate these parameters given the transition times  $\{t_m\}_M$ , we can use a typical EM algorithm designed for GMM by the following iteration formulae:

$$w_j^{(r+1)} = \frac{1}{n} \sum_{m=1}^M P(j|t_m) \quad (5)$$

$$\mu_j^{(r+1)} = \frac{\sum_{m=1}^M P(j|t_m)t_m}{\sum_{m=1}^M P(j|t_m)} \quad (6)$$

$$\sigma_j^{2(r+1)} = \frac{\sum_{m=1}^M P(j|t_m)(t_m - \mu_j^{(r+1)})^2}{\sum_{m=1}^M P(j|t_m)} \quad (7)$$

where

$$P(j|t_m) = \frac{w_j^{(r)} p(t_m|j; \mu_j^{(r)}, \sigma_j^{(r)})}{\sum_{k=1}^K w_k^{(r)} p(t_m|k; \mu_k^{(r)}, \sigma_k^{(r)})}. \quad (8)$$

The global transition-time distribution parameterized by  $\{w_j, \mu_j, \sigma_j^2\}_K$  discovered in the M step will be propagated into the E step of the next EM iteration. After a few iterations, the EM algorithm will converge to the maxima of  $p(\Theta|Z)$ . The proposed EM algorithm for traffic state monitoring within a time-window is shown in Figure 5.

### 3.4. Initialization

In this paper, we divide the time-line into many overlapped time-windows for the handling dynamic

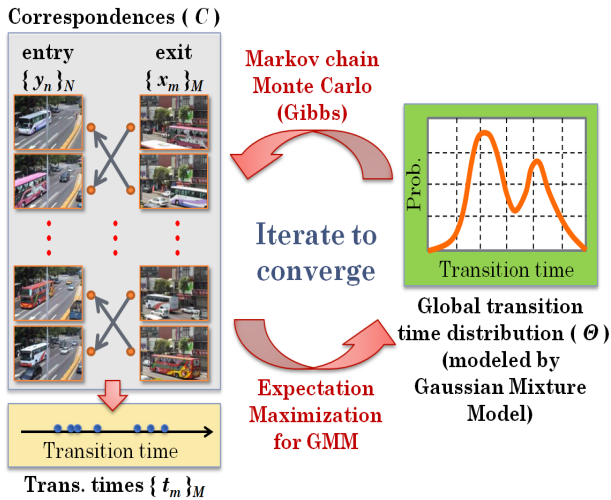


Figure 5: Proposed Expectation–Maximization algorithm for simultaneously estimating object correspondence and the parameters of the transition time distribution.

changes of the traffic flow, as shown in Figure 4, and apply the proposed EM algorithm to every time–window. An important step for performing the proposed EM algorithm in each time–window is the initial setting. For the first time–window, we enter the EM by initializing the global transition–time distribution  $\Theta$  with the Gaussian mixture model of  $K$  equal weighting, broadly separated, and wide–bandwidth Gaussians. We assume this initialization of GMM can cover most regions of the transition time. Furthermore, for the initial setting of the following time–windows, since they are partially overlapped with the previous ones, we would like to propagate the correspondences of the overlapped subsections from the previous window into the current window and use them as the initial conditions. For the data in the non–overlapped subsection which is new to the last time window, we just randomly assign the correspondences of this period. That is, we enter the EM by initializing the correspondences  $C$  which consist of the propagated and randomly assigned correspondences. This method for initialization can help the algorithm to maintain some information from the previous time–window, while still have the ability to cover the message from the new data.



Figure 6: The experimental environment: monitor the traffic between two non–overlapping FOVs at Mackay Memorial Hospital and the NTHU flyer.

#### 4. EXPERIMENTATION AND RESULTS

Our experimental environment is shown in Figure 6. We mounted two cameras at the Mackay Memorial Hospital and the overpass in front of National Tsing Hua University. In Figure 6, we can see that the FOVs of these two cameras are non–overlapped and these two scenes are linked by Kuang Fu Road (noted by the red line). Besides, there are three intersections and some bus stops on the road between these two FOVs. This makes the traffic more complicated. The traffic videos were taken from about 9:45 in the morning till 18:50 in the afternoon. We divide the time–line into 37 overlapped time–windows. The length of each time–window is 60 minutes, with 45 minutes overlapped with the previous window.

We first check the traffic videos and select frames with the buses manually. All the entry/exit times of buses are recorded. Then we apply the proposed EM algorithm for each time–window. Here we use a Gaussian mixture model with 3 Gaussians to model the transition–time distribution. For the first time–window, the parameters of the GMM are initialized with  $\mu = \{50, 150, 250\}$ ,  $\sigma^2 = \{200, 200, 200\}$  and  $w = \{0.33, 0.33, 0.33\}$ . Then the initialization for EM in all the other time–windows follows the method we described in Subsection 3.4. Figure 7 shows some examples of the transition–time distribution in different time–windows. We can see that the distribution calculated by our algorithm is reasonably close to the ground truth. To infer the traffic flow state, we can first use the average transition time of the last 15–minute observation within each time–window as a statistic to represent the state.

As shown in Figure 8, the chart of the traffic-flow state may give us the direct message about how the traffic dynamically changes over time. For instance, we can probably guess that the increase of the average transition time at about 12:00 is due to the lunch-time traffic, which is always a rush hour. Moreover, we could also express the traffic flow state by classifying the traffic changes as “stable”, “increasing”, and “decreasing,” as shown in Figure 9. This figure indicates how the traffic changes relative to the previous stage and this kind of description is much more user-friendly.

## 5. CONCLUSION

We propose an efficient method to probabilistically model the dynamic traffic flow between non-overlapping FOVs. Unlike previous works, our approach does not attempt to directly build the object correspondence across non-overlapping cameras. Instead, we model object correspondence and the parameters estimation of the transition time model as a unified problem. By building the physical connection between the transition time model and the object correspondence, the proposed EM-based framework can iteratively determine the optimal object correspondence and the model parameters. In addition, by dividing the time-line into many overlapped time-windows, our method can sequentially infer the time-varying traffic flow and recognize the dynamic changes of the traffic status over time. Moreover, our system is efficient and may provide a new thinking to well utilize the existing surveillance cameras for wide-area traffic monitoring. The experiments have shown that our approach performs well in a complicated traffic environment in real life.

## 6. ACKNOWLEDGEMENT

This work was partly supported by the National Science Council of Taiwan, R.O.C. under grant number NSC 96-2221-E-009-125-MY2 and NSC 97-2221-E-009-132-MY3.

## REFERENCES

- [1] A.D. May, “Traffic flow fundamentals,” *Prentice Hall*, 1990.
- [2] B. Fishbain, I. Ideses, D. Mahalel, and L. Yaroslavsky, “Real-time vision-based traffic flow measurements and incident detection,” *Proceedings of SPIE*, Vol. 7244, 2009.
- [3] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, “A survey of video processing techniques for traffic applications,” *Image and Vision Computing*, Vol. 21, pp. 359–381, 2003.
- [4] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, “Tracking across multiple cameras with disjoint views,” In *IEEE International Conference on Computer Vision*, Vol. 21, pp. 952–957, 2003.
- [5] O. Javed, K. Shafique, Z. Rasheed, and M. Shah, “Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views,” *Computer Vision and Image Understanding*, Vol. 109, pp. 146–162, 2008.
- [6] B. Song, and AK Roy-Chowdhury, “Stochastic adaptive tracking in a camera network,” In *IEEE International Conference on Computer Vision*, pp. 1–8, 2007.
- [7] A. Rahimi, B. Dunagan, and T. Darrell, “Simultaneous calibration and tracking with a network of non-overlapping sensors,” In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, 2004.
- [8] Y. Sheikh, X. Li, and M. Shah, “Trajectory Association across Non-overlapping Moving Cameras in Planar Scenes,” In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–7, 2007.
- [9] D. Makris, T. Ellis, and J. Black, “Bridging the gaps between cameras,” In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. II-205-II-210, 2004.
- [10] C. Stauffer, “Learning to track objects through unobserved regions,” In *IEEE Computer Society Workshop on Motion and Video Computing*, Vol. 2, 2005.
- [11] K. Tieu, G. Dalley, and WEL Grimson, “Inference of non-overlapping camera network topology by measuring statistical dependence,” In *IEEE International Conference on Computer Vision*, Vol. 2, 2005.

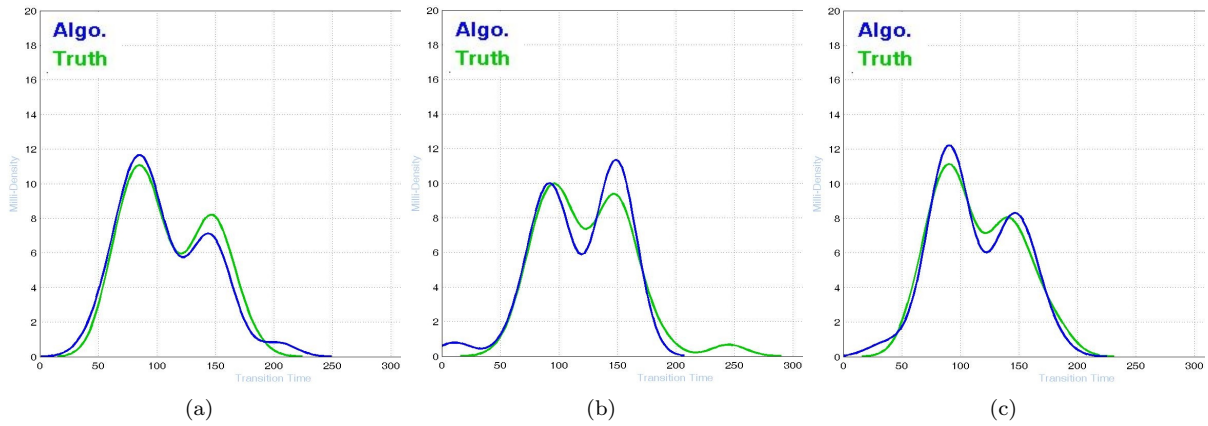


Figure 7: The transition time distribution in 3 different time-windows ( $x$ -axis: transition time;  $y$ -axis: Probability). (a)Time-window 10 : 45 ~ 11 : 45. (b)Time-window 13 : 45 ~ 14 : 45. (c)Time-window 14 : 30 ~ 15 : 30.

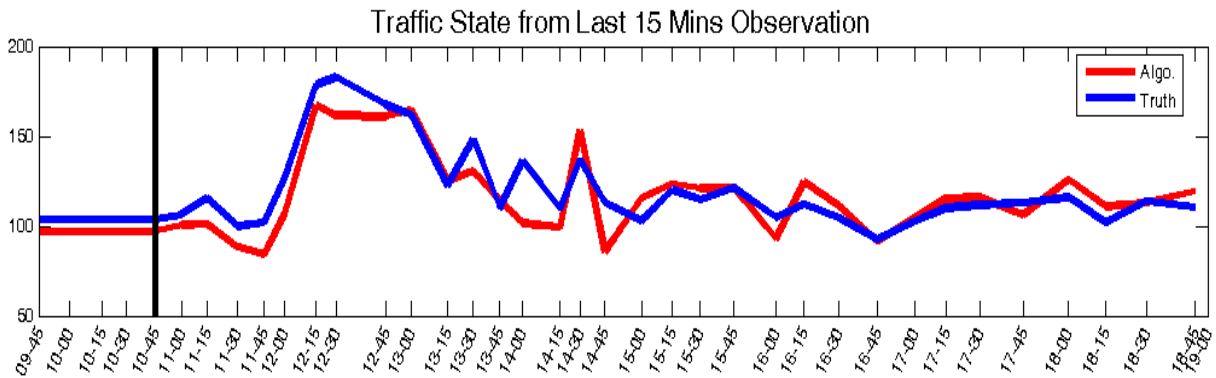


Figure 8: Traffic flow state expressed by the average transition time of the last 15-minute observation within each time-window. The red line is computed by our proposed method; The blue line is from the ground truth.

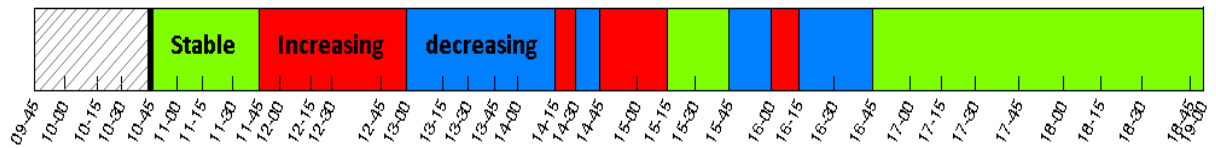


Figure 9: Express the traffic flow state by classifying the traffic changes as “stable”, “increasing”, and “decreasing.” (Green color: stable traffic; Red color: increasing traffic flow; Blue color: decreasing traffic flow).