Traffic Congestion Estimation Using HMM Models Without Vehicle Tracking

Fatih Porikli and Xiaokun Li

***Abstract*— We propose an unsupervised, low-latency traffic congestion estimation algorithm that operates on the MPEG video data. We extract congestion features directly in the compressed domain, and employ Gaussian Mixture Hidden Markov Models (GM-HMM) to detect traffic condition. First, we construct a multi-dimensional feature vector from the parsed DCT coefficients and motion vectors. Then, we train a set of left-to-right HMM chains corresponding to five traffic patterns (empty, open flow, mild congestion, heavy congestion, and stopped), and use a Maximum Likelihood (ML) criterion to determine the state from the outputs of the separate HMM chains. We calculate a confidence score to assess the reliability of the detection results. The proposed method is computationally efficient and modular. Our tests prove that the feature vector is invariant to different illumination conditions, e.g. sunny, cloudy, dark. Furthermore, we do not need to impose different models for different camera setups, thus we significantly reduce the system initialization workload and improve its adaptability. Experimental results show that the precision rate of the presented algorithm is very high around 95%.**

I. INTRODUCTION

Although the dominant technology for current vehicle traffic management systems is loop detectors, which are buried underneath highways to count vehicles passing over them, video monitoring systems promise more advantages [1]. First, more traffic parameters can be estimated into the system. Second, cameras are less disruptive and less costly to install than the loop detectors and other pneumatic sensors. Third, vision based systems provide more precise information than the nets of loop detectors. Therefore, video cameras increasingly becoming more popular in traffic monitoring and control systems.

An efficient traffic management system needs accurate traffic condition information. Since a modest system may consists of hundreds of video cameras, the computational complexity is another important consideration. Such systems also require maximum automation to decrease the burden on human operators as well.

Most existing vision systems for monitoring road traffic relied on stationary cameras and vehicle tracking, [2], [4], [6], [12], [1], and [13]. Sullivan [4] set up a system that can locate and track vehicles in 3D space when they move across the ground plane, classify their trajectories, and take account of occlusions of vehicles by stationary parts of the scene or occlusions between vehicles. Malik [11] proposed an occlusion reasoning for robust multiple car tracking. Koller [12] employed a contour tracker and affine motion model based Kalman filters to extract vehicle trajectories and used a dynamic belief network to make inferences to traffic events happened on highway. One main disadvantage of the tracking based systems is that their accuracy relies on the tracking performance. Depending on the lighting conditions, speed of the traffic, and object occlusion, the tracking can become unstable easily. Cucchiara [7] presented a system for detecting vehicles in urban traffic scenes by means of rule-based reasoning on visual data. Six traffic events were defined and tested in their system. Shuming [8] used a non-parameter regression method to forecast traffic incident from signal curve extracted from moving area. Maurin [9] designed a system which addresses a multi-level approach to monitoring traffic scenes using the technologies of optical flow, Kalman filtering, and blob merging. Yu [10] proposed a tracking based algorithm that extracts traffic information from compressed video and uses the ratio between the moving blocks and all blocks to estimate traffic situation. However most of these systems are designed for specific camera setups and computationally expensive.

In this paper, we propose an accurate, computationally simple, lighting and camera setup independent method. Instead of monitoring traffic by tracking individual vehicles, we construct motion and residual features for separate lanes and build event models using them. Since most traffic videos are already encoded using the MPEG compression, our discussion is focused on the MPEG video data.

Compressed domain analysis have significant advantages. Unlike the pixel domain techniques that require decoding of the entire input video before event detection, it is computationally inexpensive. Furthermore, compressed video contains useful information such as color, texture, edge and other spatial frequency statistics. Most importantly, compressed video embodies valuable motion information in terms of block-based motion vectors. We fuse the motion, color, texture, and other information embedded in compressed data to construct a multi-dimensional feature vector for a group of frames. We define a traffic event as a stochastic temporal process such that its features at multiple temporal scales are the samples of the stochastic process to construct an empirical distribution associated with the event. Since HMM's effectively capture the dynamic properties of the stochastic processes and successfully represent temporal continuity, we use a set of mixture of HMM's to model

Fatih Porikli is with the Audio/Video Content Analysis Group, Mitsubishi Electric Research Laboratories, Cambridge MA 02139, USA fatih@merl.com

Xiaokun Li is with the Department of Electrical and Computer Engineering, University of Cincinnati, Cincinnati, OH, 45221, USA lixiaoku@ececs.uc.edu

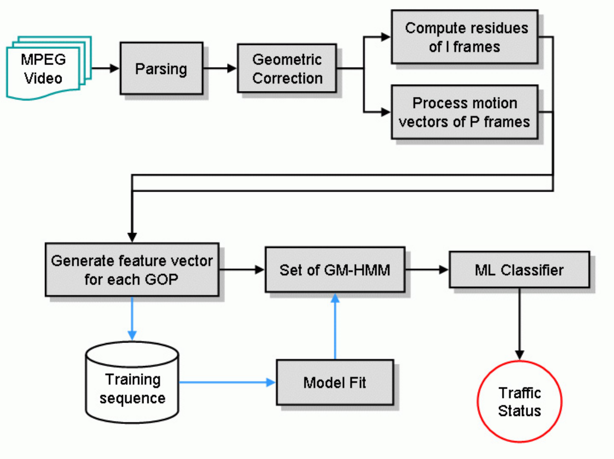


Fig. 1. Flow Diagram of congestion detection.

traffic events. Since the feature values are continuous, we employ Gaussian shape functions. The proposed method has four main stages as shown in fig. 1:

- Parsing
- Feature vector extraction
- Off-line GMHMM training
- Real-time Maximum Likelihood classification

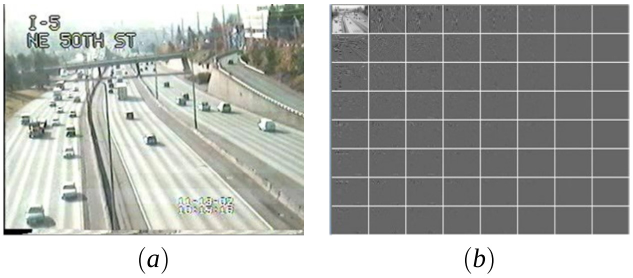
In section II, we describe the parsing process and feature extraction. We explain training of GM-HMMs and classification using ML in section III. We present the experiment results in Section IV.

II. MPEG Parser and Feature Extraction

MPEG compression scheme reduces the spatial redundancy in one frame by using Discrete Cosine Transform (DCT) and temporal redundancy between successive frames via motion compensation to achieve a low-bit rate compression. The result of motion compensation is stored as motion vector in video. An MPEG video consists of a sequence of intra-coded I frames with a number of B and P frames, where a P frame is predicted from the immediately preceding I or P frame, and a B frame is bidirectionally interpolated using the two I or P frames before and after it. The basic unit of a sequence is group of pictures (GOP) and its typical encoding order is I B B P B B P B B P. We only use the I and P frames because the B frame information is already contained within the I and P frames. Compressed video encodes an I frame using the DCT coefficients C_{uv} of a $N \times N$ image region $\{I_{xy} : 0 \leq x \leq N, 0 \leq y \leq N\}$

$$C_{uv} = \frac{1}{N}K^2 \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} I_{xy} \cos\frac{\pi u(2x+1)}{2N} \cos\frac{\pi v(2y+1)}{2N} \qquad (1)$$

where u and v are the horizontal and vertical frequencies $(u,v = 0,..,N-1)$, $N = 8$, and $K = \frac{1}{\sqrt{2}}$. When $u,v = 0$, this coefficient is called as the DC parameter $(DC = C_{00})$ and it is considered as a color mean. The remaining coefficients $u,v = 1,...,N-1$, called as AC, describe the spatial frequency energy and directionality. Since the most prominent texture and edge information is captured in the lower indexed AC terms, we use lower indexed DCT coefficients to compute an AC mean value



(a) (b)

Fig. 2. (a) An I frame from 352×240 input video (b) magnitude of the 8×8 DCT block matrix. Upper left is the DC values.

$$\overline{AC} = \frac{1}{K}\left(\sum_{u=1}^{K} C_{u0} + \sum_{v=1}^{K} C_{0v}\right). \qquad (2)$$

Note that the DC parameter and AC mean value only exist for the I frames. We compute these features using the Y color channel since illumination has the highest resolution in the MPEG compression (MPEG uses the YUV color space). The DC and AC coefficients of a sample I frame are given in fig. 2.

Motion vectors (MV) only exist in P and B frames. There is one motion vector for each block. Motion vectors are obtained by searching for the similar block. Although motion vectors represent the best color match instead of the true motion, they still give important motion information. Motion vectors that have nonzero value indicate a moving object in the spatial domain. The average direction of the majority of the motion vectors reflects the global motion, and the average magnitude of the motion vectors indicates the average velocity within the frame. To obtain more reliable motion information, a trimmed mean filter is employed to remove the marginal values as

$$mv_{ij} = \frac{1}{7}\sum_{m=1}^{7} mv^{*}_{ij} \qquad (3)$$

where mv^{*}_{ij} is the magnitude ordered MV's $|mv_0| \geq |mv_1| \geq \ldots \geq |mv_9|$ within the 8-neighborhood of block (i,j) including the center block (i,j). This is simple but still effective pruning technique.

A predefined region of interest (ROI) mask is applied to the DCT coefficients and MV's. A ROI corresponds to a traffic lane. Since traffic camera is assumed to be stationary, these regions are entered once at the beginning. We use only the DCT coefficients and MV's within the corresponding ROI's and dismiss other coefficients when we compute the feature vector. To make the feature vector invariant to different camera setups, we apply a geometrical correction. First, an affine transformation with rotation matrix R and a translation vector T maps the block locations onto a reference coordinate system $p_r = Rp + T$ where $p : (i,j)$

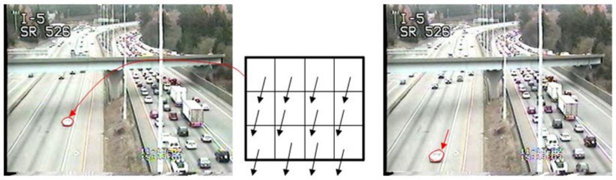


Fig. 3. Relationship of motion vectors and moving objects. MV's reflect the marked moving object between the successive frames.

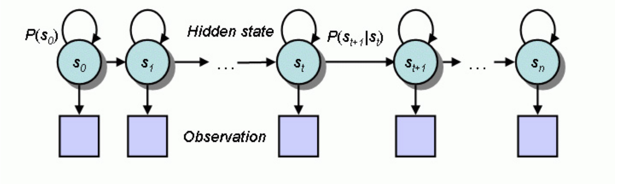


Fig. 4. Left-to-right HMM topology is used for continuous processes.

is the original coordinate of the block. The rotation and translation matrices are obtained at the initialization stage using the shape of the ROI. After this transformation, an area compensation along the axis of camera coordinate system is employed. This compensation assigns the weight parameters to the blocks within the ROI such that the contribution of blocks are proportional with respect to their distance from the camera imaging plane. For instance, the MV's of two blocks that show closer and faraway areas of the road will become identical in case of the traffic flow speed remains same on the corresponding parts of the road. Note that, before this compensation, the motion vectors have different magnitudes since vehicles will appear smaller in distance, thus their motion vectors.

A. Feature Vector

Traffic congestion is defined by two important property; speed and density of the traffic. Therefore, we designed our feature vector such that it captures the speed and density of vehicles.

For each ROI of a GOP, a single feature vector is constructed. One component of this vector represents the average difference of the DC parameters and it indicates the density and speed of the traffic. It becomes large for higher speeds and larger number of vehicles. The average DC difference R_{dc} is the ratio of the residues of DC components, which are parsed from two consecutive I-frames

$$R_{dc} = \frac{M_{dc}}{M} \sum_{i \in ROI}^{M} (DC_{i,j,t} - DC_{i,j,t-1}) \tag{4}$$

where M_{dc} is the number of blocks whose residue is larger than zero and M is the total number of blocks in the corresponding ROI. Similarly, the second component is the moving AC difference R_{ac}, which is the ratio of the residues of AC means

$$R_{ac} = \frac{M_{ac}}{M} \sum_{i \in ROI}^{M} (\bar{A}C_{i,j,t} - \bar{A}C_{i,j,t-1}) \tag{5}$$

where M_{ac} is number of blocks whose AC residue is larger than zero.

Our experiments show that the speed may also represented by the motion vectors statistics. The mean μ_{mv} and variance σ_{mv} of MV's are calculated within an ROI for the all P frames a GOP (usually a total of 4 P frames) using the trimmed MV's. Finally, $h_{mv} m_{mv} l_{mv}$ are the number of the MV's into three magnitude bands high, middle, and low. These features represent the distribution of the speed within the ROI. Since a GOP contains multiple P frames, the mean is assigned for each band. Then, the seven-dimensional feature vector is constructed as $v = [R_{ac} R_{dc} \mu_{mv} \sigma_{mv} h_{mv} m_{mv} l_{mv}]$. One advantage of this feature vector is that the motion energy, which changes along temporal direction, is accurately described. Furthermore, since all components in vector are density parameters, the vector is invariant to the size of inspected area. Another useful property is the values of these density parameters are insensitive to the different illumination since the residual is used.

III. GM-HMM

The normal traffic situation can be roughly categorized into two states, open and congestion. But we observed that such a classification is not enough to describe the traffic situation. Thus, in our system we used five traffic patterns; Stopped (S), Heavy congestion (HC), Mild congestion (MC), Open flow (OF), and Empty(E) are defined. Stopped: there is a large number of vehicles and almost all of the vehicles run very slowly or completely stopped. Heavy congestion: there is a large number of vehicles and most vehicles run slowly, Mild congestion: most of the vehicles run at half speed. Open flow: vehicles run at normal speed. Empty: there is no vehicle or minimum number of vehicles in the ROI.

An HMM is a probabilistic model composed of a number of interconnected states a directed graph, each of which emits an observable output. Each state is characterized by two probability distributions: the transition distribution over states and the emission distribution over the output symbols. A random source described by such a model generates a sequence of output symbols as follows: at each time step the source is in one state, and after emitting an output symbol according to the emission distribution of the current state, the source jumps to a next state according to the transition distribution of its current state. Since the activity of the source is observed indirectly, through the sequence of output symbols, and the sequence of states is not directly observable, the states are said to be hidden. Since traffic event is a continuous process and the profile of the probability density function in one state is a combination of several Gaussian curves, an HMM with Gaussian mixtures is selected to model traffic event. The parameters of the HMM is denoted as $\lambda = \{A, B, \pi\}$. Here, the initial state distribution is given

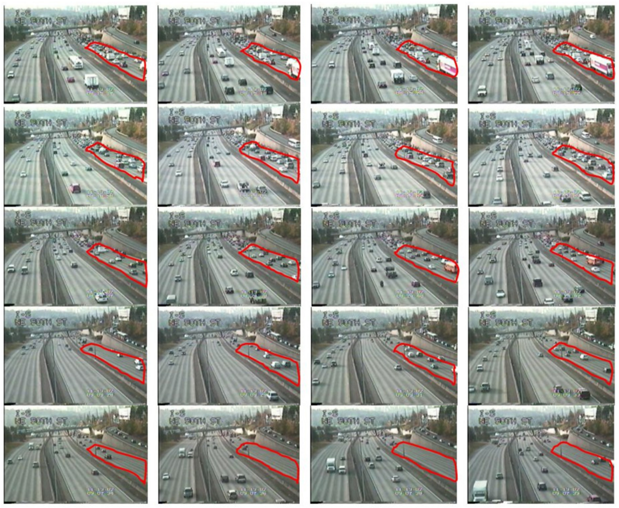


Fig. 5. Each row illustrates one of the five traffic events happened in the marked ROI. From top to bottom: the stopped, heavy, light, open, and empty traffic states. Each column corresponds to consecutive frames, i.e. initial, and 50, 100, 150 frames apart.(courtesy of DoT, Washington State)

by $\pi = \{\pi_1,..,\pi_H\}$ where $\pi_i = P(q_1 = i)$, and H is the number of hidden states, $q_1 = i$ is the i state at $t = 1$. State transition matrix is represented as

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1H} \\ . & . & . \\ a_{H1} & \cdots & a_{HH} \end{bmatrix} \tag{6}$$

where $a_{ij} = P(q_{t+1} = j|q_t = i)$. The observation probability distribution is denoted as $B = [b_1(v),..,b_H(v)]$ where in state j: $b_i(v) = P(v_t|q_t = i)$,

$$b_i(v) = \frac{1}{(2\pi)^2 \det(\Sigma)} e^{-\frac{1}{2}(v-\mu_t)^t \Sigma^{-1}(v-\mu_t)} \tag{7}$$

where v is feature vector, and Σ is the correlation vector.

The above unknown GM-HMM parameters are learned by use of Expectation Maximization (EM) algorithm. The EM algorithms perform an iterative computation of maximum likelihood estimation when the observed data are incomplete. The aim of parameter learning is to find the model parameter λ which maximizes $\lambda = \arg\max(\log p(v|\lambda))$ for a given feature vector v. One EM algorithm, Baum-Welsh algorithm, is applied to learn the traffic event model parameters. The learning process produces a sequence of estimates for *lambda*. After setting the initial values, the parameter estimation is repeated until the reaches to a local maximum. One advantage of the EM algorithm is the convergence is guaranteed and the convergence time is short (usually less than 10 times in our system). Also, the local maximum is usually an adequate model for the data. The left-to-right model is chosen as the HMM topology for all states due to its simplicity. One advantage of it is the model associates time with model states in a fairly straightforward manner. Fig. 4 illustrates the left-to-right topology.

Five identical topology GM-HMMs are trained for the five traffic states using the hand-annotated training data including different camera setups and illumination conditions. Then, the ML-based classifier is designed to detect the traffic event. Fig. 6 illustrates the classifier. The input sequence of feature vectors is fed into the trained GM-HMMs and then, the most likely sequence of states and the corresponding likelihood to each class is determined using the Trillis algorithm (an alternative is the Viterbi algorithm). Finally, the class with the highest likelihood is picked up to determine the traffic event. In our system, the Trillis calculation is implemented.

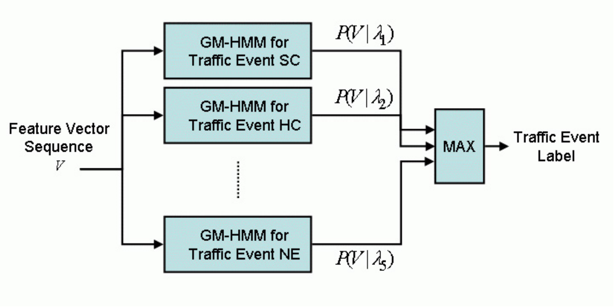


Fig. 6. GM-HMM classifier.

Since the traffic classification is not a completely objective process, it will be very helpful that the system can output the traffic detection result with a confidence score. The confidence score should be low in case of erroneous and large in case of correct detection of the state. The distance between the highest likelihood and the second highest likelihood fits this requirements. Another possible confidence score definition is the value of the highest likelihood itself. Still another measure is the transition between the previous detection result and the current state.

IV. EXPERIMENTAL RESULTS

We evaluated the proposed method using different data sources of real traffic scenes provided by the Department of Transportation, Washington State. The data set includes various illumination conditions, e.g. sunny, overcast, dark, nighttime. A total of 600 minutes, which corresponds to interstate highways, are chosen for testing. All testing clips are hand-labeled to make a comparison with a ground truth. The training video data is chosen such that there is no overlap with the testing data. The total length of the training data for each of the five traffic states is around 15 minutes.

Fig. 7-a shows a sample output of the feature extraction. The results of the traffic taken from US Highway I-5 NE are shown for 4000 GOPs (about 33 minutes) are presented. The ROI is the most left lane. All of the five states exist in this clip. From top to bottom, are the DC difference, AC difference, mean of MV's, variance of MV's, means of the MV's in high, middle, and low bands. Fig. 7-b presents the estimation result. The first row is the confidence score calculated using the distance between the highest likelihood and the second highest likelihood, the second row is unfiltered direct output from ML classification, the

$sequence$	$2-states$	$4-states$	$5-states$
$Source-1$ (I5NE, 33min)	86%	90%	95%
$Source-2$ (I5SR, 50min)	84%	91%	96%
$Source-3$ (I405, 40min)	81%	90%	94%
$Source-4$ (SR520, 40min)	83%	92%	97%

TABLE I

ACCURACY RATE FOR 2-, 4-, AND 5-STATES FOR TEST DATA THAT CONTAINS DIFFERENT LIGHTING CONDITIONS AND CAMERA SETUPS.

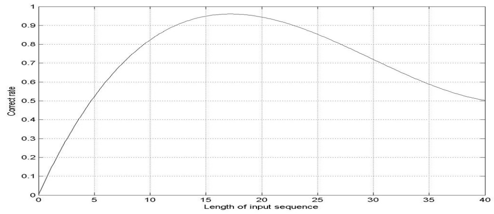


Fig. 8. Optimum length of input sequence.

third row is the result after time-wise median filtering, and the last row is the hand-labeled result .

It is visible that all of the existing traffic states are successfully detected. We compared our results with the hand labeled ground-truth. When we examine the 'false' alarms given by the ML, another interesting fact is found that our system is more sensitive than the human operator. We saw that most false alarms, the traffic state changes very rapidly, which is very demanding for human operator to find, and the state that indicated by our estimator is the correct one. Even assuming all false alarms as real false detection and directly comparing the hand-labeled ground truth with the initial output from ML, the accuracy rate reaches 94%. The accuracy rate is defined as the ratio of correctly estimated states to the total length of the data. The correct rate raises 97% for the filtered result, i.e. only 3% error rate. In case of slowly state changing traffic, the accuracy rate improves to 98% for the initial output and 99% for the filtered result. Fig. 7-b also shows the fitness of the confidence score. We find that the score is always low at the place of event change and the false alarm.

Since for each ROI of a GOP a single feature vector is constructed, the computational complexity is very low. In case of an ordinary MPEG sequence that has 2 GOP per second and a traffic setup that consists of 3 ROI's, we need to compute only 6 feature vectors at a second. We find that our algorithm can process at least six video encoded streams simultaneously in real-time.

We also implemented different state number of states; 2-states scheme (open traffic and congested traffic) and 4-states scheme (stopped, heavy, light, and open). The comparison of these 2-, 4-, 5-states schemes is given in table I. We observed that increasing the number of states is not possible due to the limited image resolution and subjective assignment of the traffic condition by the human operator. With the higher number of states (more than 6), the accuracy decreases rapidly.

The optimum length of input feature sequence is also learned by testing different length of input sequence (the number of feature vector that are fed into the GM-HMM in detection process). We exhaustively tested the various input lengths. Fig. 8 illustrates the optimum length of input sequence after a polynomial fit. From the graph, the range of the optimum value is obtained between 15 to 20.

V. CONCLUSIONS

We presented a traffic congestion estimation method that directly extracts features in the MPEG compressed domain.

Our system has several advantages: 1) it is highly accurate, i.e. precision rate is around 95%, 2) computational inexpensive, i.e. we can process more than 6 encoded streams real-time on a P4 3Ghz platform, 3) very agile and has a small latency approximately 2 seconds, which is the best reported result we are aware of, 4) robust towards illumination changes, 5) invariant to different camera setups. Furthermore, we provide a confidence score to assess the reliability of the estimation.

The compressed domain traffic congestion estimation method significantly improves the performance of traffic management systems by providing timely and accurate data.

REFERENCES

[1] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A Real-time computer vision system for Measuring Traffic Parameters", *CVPR*, pages 495-501, 1997

[2] S. Kamijo, Y. Matsushita, K. Ikeuchi, M. Sakauchi, "Traffic Monitoring and Accident Detection at Intersections", *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no 2, pages 108-118, 2000

[3] I. Masaki, "Machine-vision System for Intelligent Transportation: The Autoscope system", *IEEE Trans. Vehicle Technology*, vol. 40, pages 21-29, 1991

[4] G. Sullivan, "Model-based Vision for Traffic Scenes using the Ground-plane Constraint", *Real-time Computer Vision, D. Terzopoulos and C. Brown, Cambridge University Press*, 1994

[5] L. Zelnik-Manor and M. Irani, "Event-Based Analysis of Video", *CVPR*, vol. 2, pages 123-130, 2001

[6] Y.K. Jung, K.W. Lee, and Y.S. Ho, "Content-Based Event Retrieval Using Semantic Scene Interpretation for Automated Traffic Surveillance", *IEEE Trans. on Intelligent Transportation Systems*, vol. 2, no 3, pages 151-163, 2001

[7] R. Cucchiara, M. Piccardi, and P. Mello, "Image Analysis and Rule-Based Reasoning for a Traffic Monitoring System", *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no 2, pages 119-130, 2000

[8] T. Shuming, G. Xiaoyan, and W. Feiyue, "Traffic Incident Detection Algorithm Based on Non-parameter Regression", *IEEE Int. Conference on Intelligent Transportation Systems*, pages 714-719, 2002

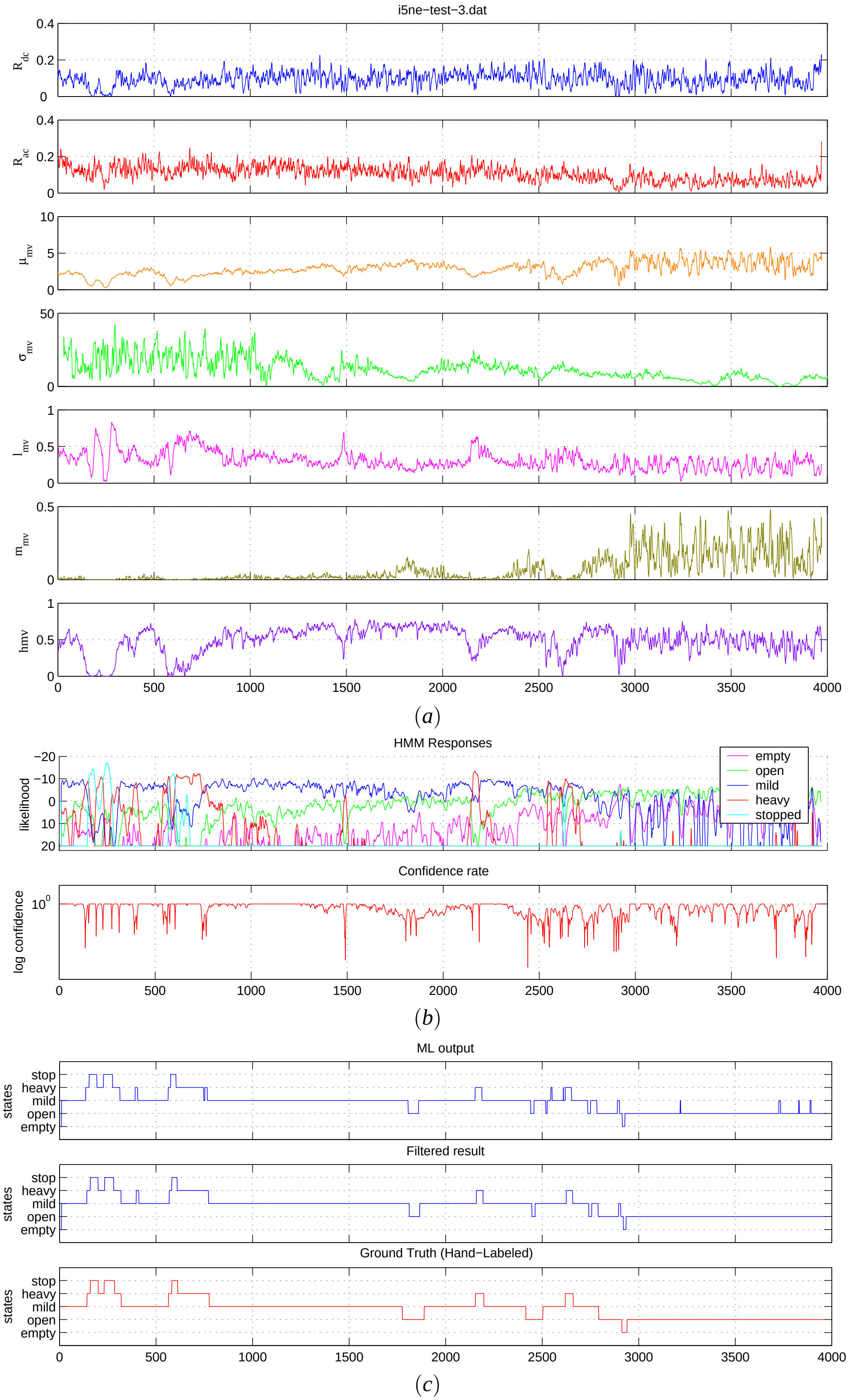
[9] B. Maurin, O. Masoud, and N. Papanikolopoulos, "Monitoring Crowded Traffic Scenes", *IEEE Int. Conference on Intelligent Transportation Systems*, pages = 19-24, 2002

[10] X.D. Yu, L.Y. Duan, and Q. Tian, "Highway Traffic Information Extraction from Skycam MPEG Video", *IEEE Int. Conference on Intelligent Transportation Systems*, pages 37-41, 2002

[11] D. Koller, J. Weber, and J. Malik, "Robust Multiple Car Tracking with Occlusion Reasoning", *ECCV*, pages 189-196, 1994

[12] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell, "Towards Robust Automatic Traffic Sence Analysis in Real-time", *ICPR*, pages 126-131, 1994

[13] A. Lipton, H. Fujiyoshi, and R. Patil, "Moving Target Classification and Tracking from Real-time Video", *WACV*, pages 8-14, 1998



(*a*)

(*b*)

(*c*)

Fig. 7. Feature extraction and detection result (a) seven-dimensional feature vector graphs, (b) up: GM-HMM responses, and down: confidence score, (c) estimated traffic condition: up: raw ML output, middle: filtered result, and down: hand-labeled ground truth for comparison. As visible, the estimator accurately detected the traffic condition with minimal delay.