Estimation of Freeway Dynamic Origin-destination Matrices: A Novel Approach

Chun-Ming TSENG [a], Yu-Chiun CHIOU [b], Lawrence W. LAN [c],

[a, b] Institute of Traffic and Transportation, National Chiao Tung University, 4F, 118 Sec.1, Chung Hsiao W. Rd., Taipei, 10012 Taiwan.
[a] E-mail: cm_tseng@motc.gov.tw
[b] E-mail: ycchiou@mail.nctu.edu.tw
[c] Ta Hwa University of Science and Technology; National Chiao Tung University, No. 1 Tahwa Rd., Chiunglin, Hsinchu, 30740 Taiwan.
[c] E-mail: lawrencelan@tust.edu.tw

Abstract: In this study, a novel approach incorporating a two-stage prediction model with an integrated algorithm is proposed to estimate freeway dynamic origin-destination (O-D) matrices. The two-stage prediction model uses growing hierarchical self-organizing map (GHSOM) to extract clusters of traffic patterns and then uses genetic programming (GP) to predict the on-ramp traffic flow in each cluster. The integrated algorithm combines cell transmission model (CTM) with extended Kalman filtering (EKF) to estimate the arrival distributions and the O-D proportions. To demonstrate the proposed approach, a field study of on-ramp traffic patterns on a freeway stretch is examined. The results show that the proposed approach can accurately predict the traffic and satisfactorily estimate the O-D proportions along the freeway stretch.

Keywords: Dynamic origin-destination, Cell transmission model, Extended Kalman filtering, Genetic programming, Growing hierarchical self-organizing map.

1. INTRODUCTION

Accurate freeway dynamic origin-destination (O-D) estimation is a prerequisite for implementing real-time traffic control applications, such as route guidance and ramp metering. Over the past two decades, some researchers have devoted to develop the estimation algorithms for the dynamic O-D matrices based on observable mainline and ramp flow rates (*e.g.* Chang and Wu, 1994; Chang and Tao, 1996, 1999; Lin and Chang, 2005, 2007) or license plate recognition (Chiou *et al*., 2012); while some others have introduced additional assumptions or exogenous information, such as route choice behaviors, prior O-D matrix information, and sequence of observational periods of traffic counts data, while estimating the dynamic O-D matrices (e.g. Bell, 1983, 1991; Yang *et al.*, 1992, 1995; Lo *et al.*, 1996; Hazelton, 2001). What seems to remain challenging in estimating the freeway dynamic O-D matrices, however, is that the number of estimated parameters is far greater than the available information.

To tackle this challenge, most previous research made assumptions on traffic patterns which may not well represent the time-of-day variation of freeway traffic conditions. For instance, Chang and Wu (1994) assumed that the vehicles entering a freeway in a specific time interval are distributed within a small range of time intervals, which certainly limits the applicability of the algorithms to other time intervals. Other studies (e.g. Chang and Wu, 1994; Chang and Tao, 1996; Lin and Chang, 2005, 2007) also made simple assumptions on traffic arrival distributions, which are not valid for elucidating the entire traffic conditions from free-flow to gridlock. More recently, Chiou *et al.* (2010) proposed an integrated algorithm which combined Daganzo's cell transmission model (1994) with extended Kalman filtering to iteratively estimate the arrival distributions and O-D proportions, respectively. However, the arrival distribution estimations assumed that the on-ramp traffic along a freeway remains unchanged over time. This assumption must be released if one wishes to predict the on-ramp traffic along a longer freeway stretch for several hours wherein the on-ramp traffic can hardly remain unchanged.

In fact, traffic patterns can change dramatically from peak hours to off-peak and vice versa. According to field observation, however, daily traffic patterns do repeat spatially and temporally over and over again. Viewed in this light, Chiou *et al.* (2013) proposed a two-stage prediction model with growing hierarchical self-organizing map (GHSOM) to extract clusters of traffic patterns and with a genetic programming (GP) to predict the traffic for each cluster separately. The accuracy of the prediction has been proven satisfactory. In this study, therefore, we further incorporate the on-ramp traffic prediction model (Chiou *et al.*, 2013) with the cell transmission model (CTM) to simulate the arrival patterns and then employ the extended Kalman filtering (EKF) to estimate the O-D proportions. To validate the proposed approach, an empirical study on a freeway stretch is examined. The remainder is organized as follows. Section 2 gives the definition of the problem, variables and related parameters. Section 3 introduces the proposed approach, including the traffic prediction model, the CTM arrival distribution model, and the EKF O-D matrices estimation. Section 4 presents a case study, followed by conclusions and suggestions for future research.

2. DEFINITIONS

A typical freeway corridor depicted in Figure 1 is considered here. The information that is readily available for estimation of dynamic O-D distribution ($b_{ij}(k)$) is the time series of entering flow $q_i(k)$, exiting flow $y_j(k)$, and mainline flow $U_l(k)$.

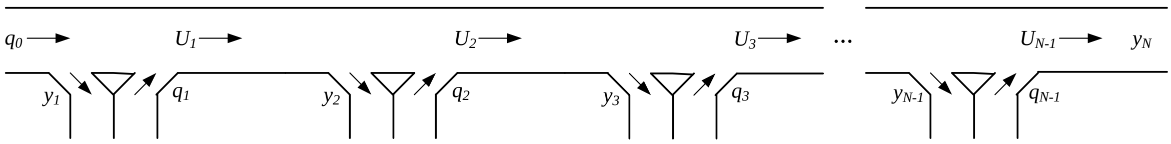


Figure 1. A typical freeway corridor

Based on Chang *et al.* (1994, 2006, 2007), the relationship between the dynamic O-D pattern, resulting link flow, and arrival pattern can be expressed by the following equations: (Make necessary revision for eqs. (1) through (10) in terms of subscripts *t*, f, F, x, and X.)

$$y_j(k) = \sum_{m=0}^{M}\sum_{i=0}^{j-1} q_i(k-m)\cdot\rho_{ij}^{m}(k)\cdot b_{ij}(k-m) \tag{1}$$

$$U_l(k) - q_l(k) = \sum_{m=0}^{M}\sum_{i=0}^{j-1}\sum_{j=l+1}^{N}\left[q_i(k-m)\rho_{il}^{m}(k)\right]b_{ij}(k-m) \tag{2}$$

where,

$y_j(k)$: the number of vehicles leaving freeway from off-ramp j during time interval *k*, *j* = 1,2,. . . , *N*-1,

$U_l(k)$: the number of vehicles crossing the upstream boundary of segment *l* during time interval *k*, l = 1, 2,. . . , *N*-1,

$q_i(k)$: the number of vehicles entering freeway from on-ramp i during time interval *k*, *i* = 1,2,. . . , *N*-1,

$b_{ij}(k)$: The proportion of $q_i(k)$ heading toward destination node *j* during time interval *k*, and

$\rho_{ij}^{m}(k)$: the fraction of $T_{ij}(k\text{-}m)$ vehicles departing from entry node *i* during time interval *k* that takes m time intervals to exiting node *j*.

Eqs. (1) and (2) express the relationship between entering traffic ($q_i(k-m)$), mainline traffic volume ($U_l(k)$-$q_l(k)$), O-D matrices proportion ($b_{ij}(k-m)$), and arrival pattern ($\rho_{ij}^{m}(k)$). Obviously, the system formulation has a large number of state parameters, *i.e.*, $b_{ij}(k)$ and $\rho_{ij}^{m}(k)$, causing low efficiency in estimation. As such, more information is required to ensure the proposed model to be computationally efficient and tractable.

3. THE PROPOSED APPROACH

3.1 The Framework

Figure 2 presents the detailed framework of the proposed approach for building a recursive dynamic O-D matrix estimation model. It uses GHSOM and GP to predict on-ramp traffic in advance, then supplies the CTM to predict arrival patterns by simulated traffic flow on freeway, and finally uses EKF to estimate O-D matrices. Specifically, in Step 1, the proposed approach inputs the rolling on-ramp traffic patterns and uses GHSOM algorithm to cluster traffic patterns. In Step 2, it matches traffic patterns into similar cluster and uses GP to predict on-ramp traffic flow. In Step 3, it determines a set of $b_{ij}(k)$ for assigning the detected on-ramp traffic to different downstream interchanges. Once the arrival distributions of all entering traffic have been simulated, $\rho_{ij}^{m}(k)$ can be computed and used to calibrate the O-D proportions of entering traffic $q_i(k)$ by EKF, namely b_{ij}'(k). Then, the new O-D proportions b_{ij}'(k) will be used to replicate a revised arrival distribution $\rho_{ij}^{m}{}'(k)$ in an iterative manner. The stop condition is set as: $\sum_i \sum_j [b_{ij}^{n-1}(k) - b_{ij}^{n}(k)]^2 < 0.05$, where *n* is the operation result of n^{th} recursive, or 500 times of recursion.

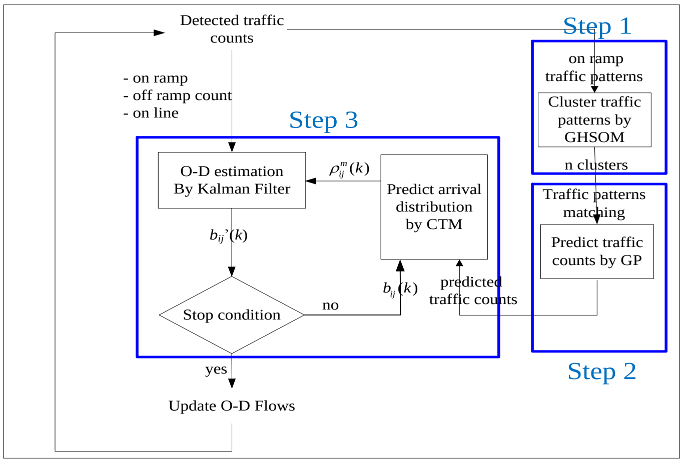


Figure 2. Framework of the proposed approach

3.2 Two-stage Traffic Prediction Model

3.2.1 Traffic patterns clustering

This study follows the two-stage traffic prediction model proposed by Chiou *et al*. (2013), which used GHSOM to identify the similar cluster of traffic patterns into appropriate different traffic patterns. The GHSOM has a hierarchical structure of multiple layers, where each layer consists of several independent growing self-organizing maps (SOM) (Rauber *et al.*, 2002). The SOM network performs learning with the following steps.

Step 1: Randomly initialize the weight vector of each neuron.

Step 2: Determine the winning neuron. The SOM network determines the winning neuron for a given input vector selected randomly from the set of all input vectors. For every neuron on the grid, its weight vector is compared with the input vector by using some similarity measures, e.g., Euclidean distance. The neuron whose weight vector is closest to the input vector is selected as the winning neuron. Eq. (3) shows how to determine the winning neuron *b*.

$$b\colon \left\| X_{kt} - w_b(l) \right\| = \min_i \left\{ \left\| X_{kt} - w_i(l) \right\| \right\} \tag{3}$$

where,

X_{kt} : a sequence of r periods traffic flow rates locating at interchange k and starting from time t and X_{kt}=[$x_k(t+1)$, $x_k(t+2)$, …, $x_k(t+r)$],

w_i : weight vector of the i^{th} neuron,

l : the number of current learning iteration, and

b : winning neuron.

Step 3: Update the weights. After a winning neuron is determined, the weight vectors of winning neuron along with its neighboring neurons are updated so as to “move” toward the input vectors according to the following equation:

$$w_i(l+1) = w_i(l) + h_{bi}(l)(X_{kt} - w_i(l)) \tag{4}$$

where,

$h_{bi}(l)$: the neighborhood function.

A widely used neighborhood function is based on the Gaussian function:

$$h_{bi}(l) = \alpha(l)\exp(-\frac{\|r_i - r_b\|^2}{2\sigma(l)^2}) \tag{5}$$

where,

$\alpha(l)$: the learning rate function which controls the amount of weight vector adjustment and decreases with the iterations,

r_i, r_b : the locations of the neuron i and winning neuron b in the lattice, and

$\sigma(l)$: the width of the neighborhood function and also decreases monotonically.

Step 4: Test the stop condition. Steps 2 and 3 are repeated until all the patterns in the training set have been processed. In addition, to achieve a better convergence toward the desired mapping, it is usually required to repeat the previous loop until some convergence criteria are met.

Based on the concept of above SOM learning process, the training algorithm of GHSOM grows in two dimensions: horizontally by increasing the size of each SOM and hierarchically by increasing the number of layers (Chiou *et al.*, 2013), stated as follows.

Horizontal growth:

Step 1: Randomly initialize the weight vector of each neuron.

Step 2: Perform the conventional SOM learning algorithm for a preset number of iterations.

Step 3: Find the error unit *e* and its most dissimilar neighbor unit *d*. The error unit *e* is the neuron with the largest deviation between its weight vector and the input vectors it represents.

Step 4: Insert a new row or a new column between *e* and *d*. The weight vectors of these new neurons are initialized as the average of their neighbors.

Step 5: Repeat steps 2-4 until the mean quantization error of the map (MQE_m) is less than ($\tau_1 \cdot mqe_u$). τ_1 is a threshold specifying the desired level of detail to be shown in a particular SOM. mqe_u is the mean quantization error of the neuron *u* in the preceding layer of the hierarchy. Eq. (6) calculates mqe_u, which is the average distance between the weight vector of neuron *u* and the input vector mapped onto this neuron:

$$mqe_u = \frac{1}{n_{C_u}} \sum_{x_j \in C_u} \|X_{kt} - w_u\|, \;\; n_{C_u} = |C_u| \tag{6}$$

where,

C_u : the set of input vectors that are mapped onto unit *u*,

w_i : the weight vector of unit *I*,

$\|X_{kt} - w_u\|$: the distance between input vector X_{kt} and weight vector w_u, and

$|C_u|$: the cardinality of the set C_u.

Furthermore, the mean of all neurons' quantization errors in the map, MQE_m, is calculated as follows:

$$MQE_m = \frac{1}{n_U} \sum_{i \in U} mqe_i, \quad n_U = |U| \tag{7}$$

where,

U : the subset of map units.

Hierarchical growth:

Step 1: Check each neuron to find out if its mqe_u is greater than ($\tau_2 \cdot mqe_0$). τ_2 is a threshold specifying the desired quality of input data representation at the end of learning process; mqe_0 is the mean quantization error of the single neuron of Layer 0. Then, assign a new SOM at a subsequent layer of the hierarchy. mqe_0 is computed as follows:

$$mqe_0 = \frac{1}{n_I} \sum_{x_j \in I} \| X_{kt} - m_0 \|, \quad n_I = |I| \tag{8}$$

where,

m_0 : the mean of the input vectors,

I : the set of the input vectors, and

mqe_0 : a measurement of the overall dissimilarity of input data.

Step 2: Train the SOM with input vectors mapped to this neuron.

3.2.2 Traffic pattern matching

Compute the representative traffic pattern (i.e. cluster seed) of each cluster produced by *GHSOM* by taking the average of all individual traffic patterns being grouping into the cluster. Then the traffic pattern matching algorithm simply assigns the input traffic pattern to a specific cluster with the most similar cluster seed based on the Euclidean distance measure. The process can be summarized as follows:

Step 0: Input a traffic pattern X_{kt}.

Step 1: Compute the cluster seed of each of clusters.

Step 2: Compute the squared Euclidean distance of each input objects, such that

$$Min(X_{kt}, X_c) = \sum_{i=1}^{r} (x_k(t+i)\text{-}x_c(i))^2 \tag{9}$$

where,

X_{kt} : input traffic pattern and,

X_c : cluster seed of cluster c.

Step 3: Assign the input pattern to specific cluster with the minimum squared Euclidean distance.

Step 4: Use the prediction patterns of the specific cluster to predict traffic flow.

3.2.3 Traffic prediction

The traffic prediction is based on the historical and primitive n time-series data. Since the number of traffic patterns needed to construct the prediction model is unknown in advance, GP algorithm is suitable for such prediction. The GP algorithm starts with a population of randomly generated individual trees; each tree corresponds to the linear combination of traffic flow. The traffic prediction contains the following steps:

Step 0: Define function set and terminal set. The function set consists of the arithmetic functions of addition, subtraction, multiplication, division, as well as a conditional branching operator. The terminal set is set as the latest *r* periods of traffic flow data.

Step 1: Initialize random population size.

Step 2: Evaluate fitness values of the trees. Randomly select trees from the population, evaluate them with training patterns belonging to this cluster, and then rank them according to their fitness values. A fitness measure is defined as follows:

$$E_{lq} = \sqrt{\frac{\sum_{i=1}^{I}\sum_{t=1}^{h}\left(x_{li}(t+r+1) - f_q(X_{li}(t))\right)^2}{h}} \tag{10}$$

where

E : fitness measure,

$f_q(\cdot)$: the mathematical expression of tree q predicting the traffic flow at next time period based on the inputted historical data at previous time periods, i.e., $f_q(X_{li}(1)) = \hat{x}_{liq}(t+r+1)$.

Step 3: If the fitness value approaches to zero, then stop the procedure. Otherwise, proceed to the next step.

Step 4: Create new individual by applying genetic operations. The genetic operations further include reproduction, crossover and mutation as follows.

Step 4-1: Reproduction. Replace the least-fit two traffic patterns by the best-fit two.

Step 4-2: Crossover. Create a new offspring by randomly combining the chosen parts of two selected trees in each parent tree and swapping the sub-tree rooted at crossover points.

Step 4-3: Mutation. Randomly select a mutation point in a tree and substitute the sub-tree rooted there with a randomly generated sub-tree.

Step 5: Generate new population by using genetic operations, and return to Step 2.

3.3 Cell-based Arrival Distribution Modeling

Following Chiou *et al.* (2010), this study employs CTM to predict the arrival distribution of an O-D pair of traffic along a freeway corridor, which is used for computing $\rho_{ij}^{m}(k)$. During free-flow periods, all vehicles in a cell can be reasonably advanced to the next cell as time evolves. One needs not to know where the vehicles are located within the cell. Namely, the system evolution obeys:

$$n_{i+1}(t+1) = n_i(t) \quad for\ t = 0, 1, 2, \ldots, T \tag{11}$$

where,

$N_i(t) - n_i(t)$ ：the amount of empty space in cell *i* at time *t*.

The traffic flow increases gradually until a queue occurs, where maximum flow and maximum number of vehicle variables are incorporated into the model.

The CTM assumes a simplified version of the fundamental diagram, typically based on a trapezium form (Figure 3) and provides simple solutions for realistic networks. It assumes a free-flow speed *v* at low densities and a backward shockwave speed *w* at high densities; both speeds are constant ($v \geq w$).

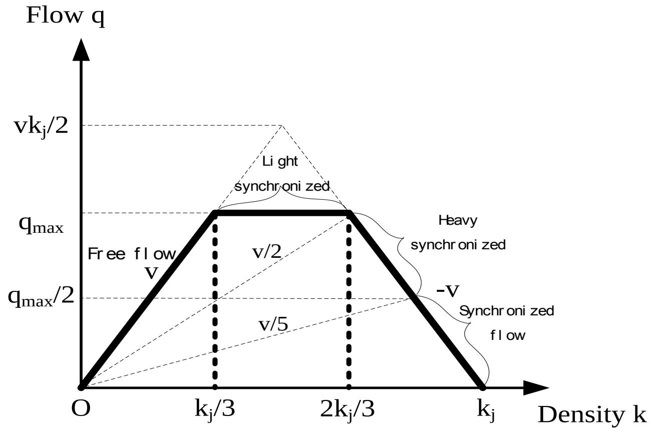


Figure 3. Fundamental diagram of CTM

The number of vehicles flowing into cell *i* within time interval *t*, $c_i(t)$, is defined as:

$$c_i(t) = \min\left\{ n_{i-1}(t),\ Q_i(t),\ \frac{w}{v}\left[N_i(t) - n_i(t)\right] \right\} \tag{12}$$

If the remained storage capacity and flow capacity in the next cell is sufficient, all vehicles will move forward into the next cell; otherwise, only a portion of them can move into the next cell according to the following logics:

$$\begin{aligned} &if \quad c_i(t+1) + q_i(t+1) \leq \min\left[Q_i, N - n_i(t+1)\right] \\ &then \quad c_{i+1}(t+1) = c_i(t+1) + r_i(t+1) \end{aligned} \tag{13}$$

$$\begin{aligned} &if \quad c_i(t+1) + r_i(t+1) > \min\left[Q_i, N - n_i(t+1)\right] \\ &then \quad c_{i+1}(t+1) = 1 - \left[\min\left[Q_i, N - n_i(t+1)\right] \Big/ c_i(t+1) + r_i(t+1) \right] \end{aligned} \tag{14}$$

3.4 Extended Kalman Filtering

The proposed estimation algorithm, based on the extended Kalman filtering concept, is as follows:

Step 0: Initialization.

Parameters settings include cell length L_i, *i = 0,1,. . . , N-1,* and time interval t_0. $\operatorname{var}\left[e(k)\right] = diag\left[r_1, r_2 ...\right]$. $X(0) = E\left[b(0)\right]$. $P(0) = Var\left[b(0)\right]$. Besides, on-ramp, link and off-ramp flows are given.

Step 1: Determine $\rho_{ij}^{m}(k)$ by CTM.

Step 2: Compute the linearized transformation matrix based on the determinant $\rho_{ij}^{m}(k)$.

$$H^{K-1} = \left[H_{rs}^{k-1} \right]$$

$$H_{j,Ni+j-i(i+1)}^{k} = \sum_{m=0}^{M} q_i(k-m) \cdot \rho_{ij}^{m}(k) \qquad for \quad 0 \le i < j \le N \tag{15}$$

$$H_{N+l,Ni+j-i(i+1)}^{k} = \sum_{m=0}^{M} q_i(k-m) \cdot \rho_{ij}^{m}(k) \qquad for \quad 0 \le i < j \le N \tag{16}$$

$$\left[H^{K-1} \right] = \left[h_1, h_2, \ldots, h_{2N-1} \right]^T$$

$$Z'(k) = \left[y_1(k), y_2(k), \ldots, y_N(k); U_1(k) - q_1(k), \ldots, U_{N-1}(k) - q_{N-1}(k) \right]^T$$

Step 3: Initialization of the sequential Kalman filtering method.

Set $b_0 = b(k+1)$

$p_0 = p_{k+1} + D$, where $D = \left[d_b, \ldots, d_b \right]$ is a covariance matrix of *W(k)*

Step 4: Sequential Kalman filtering iterations.

For *i = 1, 2, ..., 2N-1*

$$g^i = p^{i-1} h_i^T \left[h_i p^{i-1} h_i^T + r_i \right]^{-1}$$

$$p^i = p^{i-1} - g^i h_i p^{i-1}$$

$$\delta^i = y_i(k) - h_i b(k-1)$$

Truncation:

$$\alpha' = \underset{0 \le \alpha \le 1}{MAX} \left[\alpha \middle| 0 \le \left[b^{i-1} \right] + \alpha \delta^i g^i \le 1 \right]$$

Set $\left[b^i \right] = \left[b^{i-1} \right] + \alpha \delta^i g^i$

Normalization:

For *m=1, 2, ..., N-2*

$$\beta_m = \sum\nolimits_{j=m+1}^{N} b_{mj}^{i}$$

$$b_{mj}^{i} = {}^{b_{mj}^{i}} \big/ {}_{\beta_m} \qquad j=m+1,\ldots, N.$$

Step 5: Stop condition.

Check the convergence of estimated O-D proportions. If the preset stop condition (convergence level or number of iterations) has not reached, then go to Step 1. Otherwise, go to Step 6.

Step 6: Prediction of the states.

Set $p_k = p^{2N-1}$ and $\left[b(k) \right] = \left[b^{2N-1} \right]$, *k* = *k* + 1, go to Step 1.

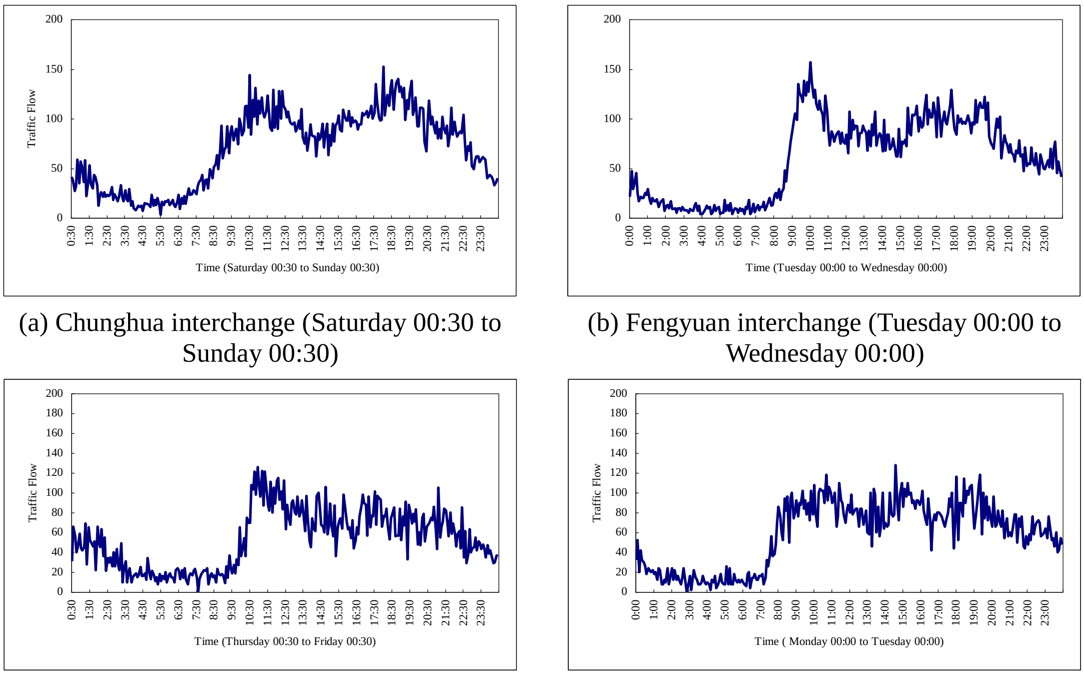
4. CASE STUDY

The 3-lane mainline freeway stretch of Taiwan Freeway No. 1 between Toufen Interchange and Beidou Interchange is studied due to its dense loop detectors. This stretch is 120 kilometers in length and has 15 on-ramp interchanges in total. An entire week (May 25-31, 2009) of 5-minute traffic counts are directly extracted from the loop detectors throughout this stretch.

4.1 Traffic Flow Prediction Models

The entire week of 5-minute traffic counts are separated into a total of 25,935 (=1,729×15) traffic patterns with a fixed length of 24 hours on a 5-minute rolling basis. These 25,935 traffic patterns are clustered by GHSOM, which results in a total of 4 layers with 36 different clusters. Note that the number of traffic patterns in each of the 36 clusters ranges from 143 (Cluster 1) to 2,488 (Cluster 6). In each cluster, we then construct a GP traffic flow prediction model based on the traffic patterns in the same cluster. Hence, a total of 36 GP traffic flow prediction models are developed.

Taking Cluster 36 as an example, it contains 488 traffic patterns during the periods of night time to next night time on weekday and weekend in the suburban areas (e.g. Chunghua interchange, Fengyuan interchange, Daya interchange and Nantun interchange) where the maximum 5-minute flow rates can exceed 150 pcu. Figure 4 illustrates four randomly selected traffic patterns, which look similar.



(a) Chunghua interchange (Saturday 00:30 to Sunday 00:30)

(b) Fengyuan interchange (Tuesday 00:00 to Wednesday 00:00)

(c) Daya interchange (Thursday 00:30 to Friday 00:30)

(d) Nantun interchange (Monday 00:00 to Tuesday 00:00)

Figure 4. Four randomly selected traffic patterns from Cluster 36 (suburban area)

This study further randomly divides these 488 traffic patterns into two sets: training set (340 patterns) and validation set (148 patterns). Based on the training patterns, the GP traffic prediction model for Cluster 36 is estimated as follows:

$$x(t+1)= 0.9967\, x(t) +1.05\times10^{-2}x(t-5)+ 1.26\times10^{-2}\, x(t-1)\, x(t-5) -1.9\times10^{-3}x(t-4)^{2}-1.08\times10^{-2}x(t-3)x(t-5)+5\times10^{-5}x(t)x(t-1)x(t-5)-4\times10^{-5}x(t-1)^{2}x(t-4)-1\times10^{-6}x(t-1)^{2}x(t-2)\, x(t-5)+3\times10^{-6}x(t-1)^{2}\, x(t-3)\, x(t-7) -2\times10^{-5}x(t-2)x(t-3)\, x(t-4)x(t-7) \quad (17)$$

where,

$x(t)$: the traffic of time t.

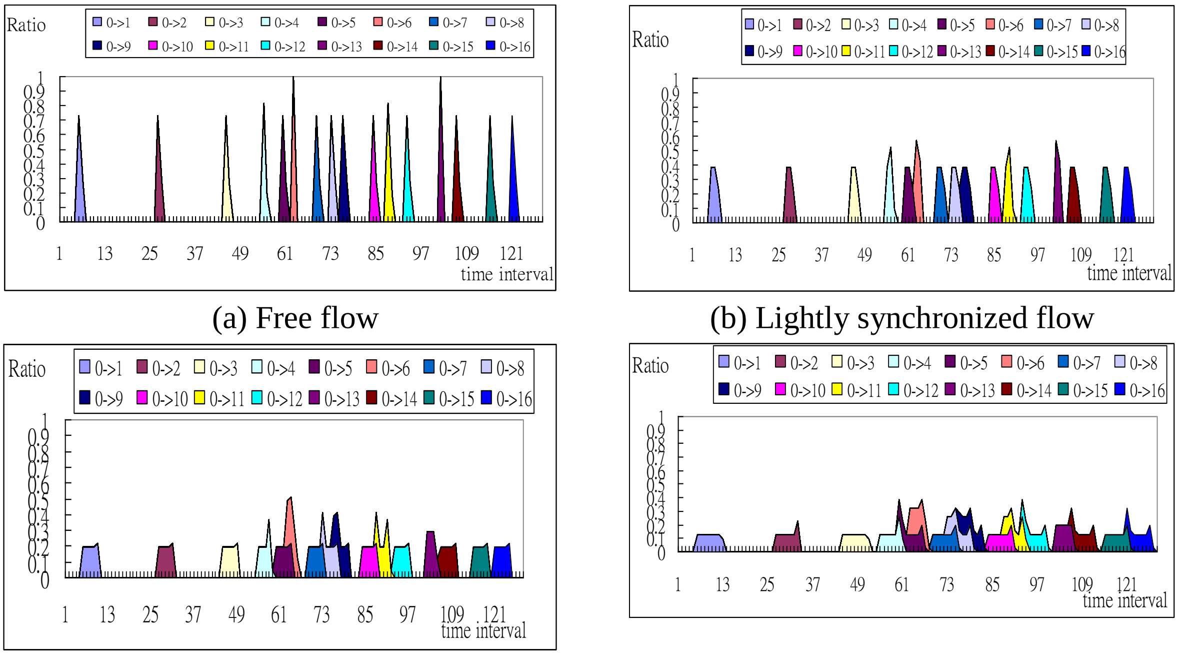
Note from Eq. (17) that the GP prediction model for Cluster 36 only needs the nearest 7 intervals of traffic to predict the very next interval of traffic. With a rolling horizon, the GP prediction model can repeatedly perform traffic flow prediction for the next 4 hours (a total of 48 time intervals) by feeding-in the newly predicted traffic flow data. The results show that the MAPE values of GP prediction model in training and in validation are 7.02% and 10.75%, respectively, which are satisfactory.

Following the same vein, the GP prediction models for the remaining 35 clusters are developed. Overall, the average MAPE values in training and in validation for the 36 GP prediction models are 6.88% and 10.35%, respectively. It indicates a satisfactory prediction of 4-hour 5-minute traffic flows within this 120-km freeway stretch.

4.2 Traffic Arrival Distributions

To show the capability of CTM in replicating the traffic hydrodynamics and to investigate the degree of traffic dispersion under various traffic conditions, simulations on this 3-lane freeway stretch are performed. The parameters are set according to the local situations: free flow speed=100 km/hr, jam density=400 vehicles per kilometer, capacity=6,000 vehicles per hour, cell storage capability=67 vehicles, time interval=6 seconds, and cell length=1/6 km.

Figure 5 shows the arrival distributions under four traffic conditions: free-flow, lightly synchronized flow, heavily synchronized flow, and congested flow. Throughout the simulations, all of the entering traffic flows are increased by the same ratio, step by step, from 10% to 100%. Under free flow condition, it is noted that almost all ranges of arrival times cover only one or two intervals. Under lightly synchronized flow condition, the remaining storage capacity and flow capacity of the next cell are sufficient; hence, all vehicles in a cell can advance into the next cell within each interval. Under heavily synchronized flow condition, the entering traffic has exceeded the remained storage capacity (i.e., flow capacity of the next cell is not sufficient); thus, only a portion of them can move forward proportionally. Under congested flow condition, it is noted that the arrival times of traffic dispersion can be as long as six to eight time intervals.



(a) Free flow

(b) Lightly synchronized flow

(c) Heavily synchronized flow

(d) Congested flow

Figure 5. Arrival distributions of entering traffic from origin 0 to various destinations (1~16)

4.3 O-D Proportions Estimation

In this study, two initial value settings are attempted: one by randomly generated (RG) technique and the other by equal share (ES) technique. Taking origin interchange No. 12 as an example, the associated O-D proportions are denoted as $b_{12,13}(k)$, $b_{12,14}(k)$, and $b_{12,15}(k)$. With RG technique, three random numbers 0.2, 0.9, and 0.5 are generated and then normalized such that the sum of three proportions equals 1; namely, $b_{12,13}$ (k)=0.125, $b_{12,14}$ (k)=0.563, and $b_{12,15}$ (k)=0.312. In contrast, with ES technique, the three proportions are simply set as $b_{12,13}$ (k)=0.333, $b_{12,14}$ (k)=0.333, and $b_{12,15}$ (k)=0.333.

The distributions of real $b_{12,15}$ proportions (from Zhanghua Interchange to Yuanlin Interchange) along with the estimated O-D proportions by both RG and ES techniques are demonstrated in Figure 6. Note that the proposed approach can predict real O-D proportions quite accurately for these two initial value setting techniques. However, RG technique seems slightly superior to the ES technique in terms of the prediction accuracy. Thus, the RG technique is adopted for predicting the remained O-D proportions.

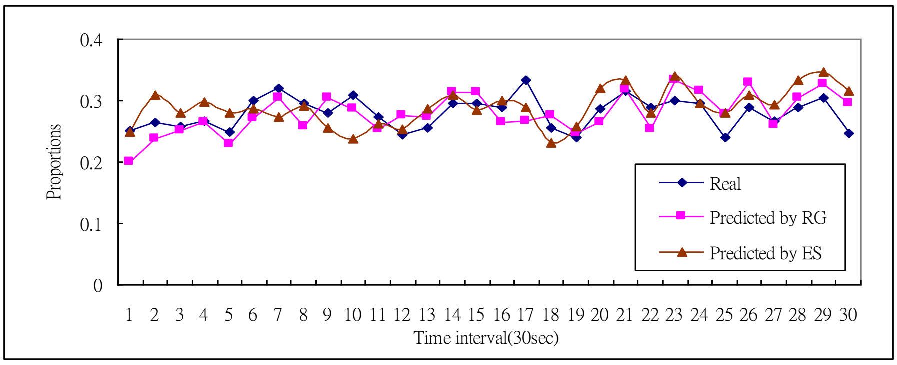


Figure 6. Comparison of real $b_{12,15}$ proportions with predicted values by RG and ES techniques

Figure 7 illustrates the convergence process for time interval k=986, $b_{1,15}$, Toufen Interchange to Beidou Interchange. The results show that the overall *RMSE* is 0.1043, indicating a rather high goodness-of-fit of the proposed approach.

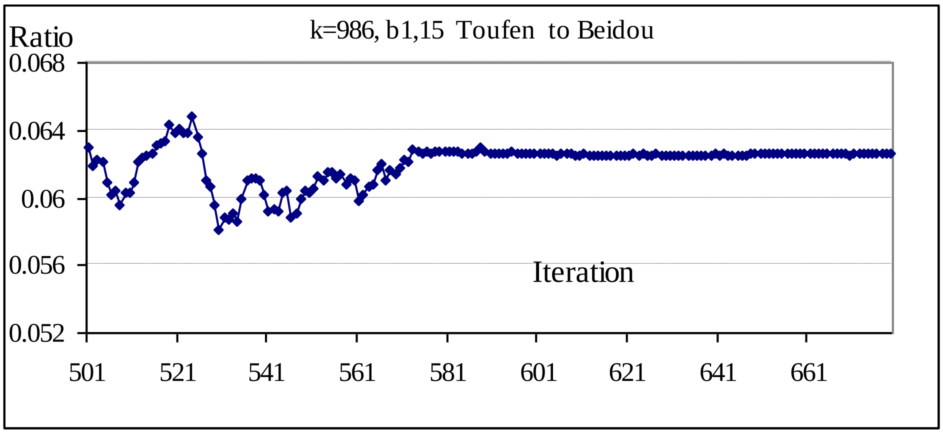


Figure 7. The convergence process

4.4 Sensitivity Analysis

Our proposed method heavily depends on the real-time fed-in traffic flow data, which are used for determining which clusters it belongs to, and furthermore, for predicting the traffic flows at the next 48 time intervals. In the previous settings, the traffic patterns have been defined as traffic flow data at consecutive 240 time intervals, or 20 hours. In the following, a sensitivity analysis with different traffic pattern lengths: 48 (4 hours), 72 (6 hours), 96 (8 hours), 120 (10 hours), 144 (12 hours), 192 (18 hours) and 240 (20 hours) time intervals is further conducted. The MAPE values for training and validation are presented in Table 1. We note that shorter lengths (*e.g.*, L=48 and 72) have relatively lower prediction accuracy than longer ones, suggesting the necessity to input a sufficient long traffic pattern for both pattern recognition and prediction. It is also interesting to note that there are no significant changes in prediction accuracy once the length of traffic patterns is longer than 120 time intervals.

Table 1 The MAPE values with different traffic pattern lengths

Lengths	Training	Validation
48	7.29%	19.72%
72	7.78%	14.74%
96	5.32%	12.01%
120	5.72%	10.34%
144	5.91%	10.86%
192	5.63%	10.38%
240	4.58%	10.07%

4.5 Comparison

To show the superior performance of the proposed method, a commonly-used traffic prediction model—the autoregressive integrated moving average (ARIMA) model is further developed for comparison. Following the same data basis as the proposed method, the ARIMA model is also developed on the previous 240 time intervals and predicts the following 48 time intervals. The *MAPE* values of training and validation datasets are 21.77% and 28.65%, respectively, which are much higher than those of our proposed method, which are 5.10% and 10.15%, respectively.

With the self-structured traffic patterns, a simplified prediction model was to be naively developed by averaging the traffic flow data at the last 48 time intervals. For example, if one traffic pattern is of interest in Cluster 1 where 143 traffic patterns have been identified, then the traffic flows at the next 48 time intervals are predicted by taking the average traffic flow of 143 traffic patterns. The MAPE values of training and validation datasets for all clusters with this simplified model are 25.21% and 32.73%, respectively, which are much higher than those of our proposed method (4.58% and 10.07%). Again, this comparison further confirms the superiority of the proposed method and it suggests the necessity of GP model.

5. CONCLUDING REMARKS

This paper contributed to propose a novel approach for estimating the freeway dynamic O-D matrices, based on a growing hierarchical self-organizing map and genetic programming, in conjunction with an integrated cell transmission model and extended Kalman filtering to iteratively estimate the arrival distributions and the O-D proportions, respectively. The case study on Taiwan's freeway has shown that the proposed approach can accurately predict the traffic and estimate the O-D proportions with rather low *RMSE*, indicating its practical applicability.

Several directions for future study can be identified. First, in the case study the O-D matrices are generated by a traffic simulation software－DynaTaiwan based on the detected traffic flows. With advanced traffic surveillance technologies, it is possible in the future to collect real-time traffic information to further validate the proposed approach. Second, the proposed approach is developed only for a freeway corridor. In the future study, route choice behaviors can be incorporated into the proposed approach to extend the applications to a freeway network. Last but not least, a comparison of the proposed approach can be made with other existent O-D estimation algorithms to demonstrate the superiority of different algorithms.

ACKNOWLEDGEMRNTS

This study was partially sponsored by Taiwan's National Science Council under contract number NSC 96-2628-E-009-171-MY3. Three reviewers' constructive comments and suggestions are highly appreciated.

REFERENCES

Bell, M.G.H. (1983) The estimation of an origin-destination matrix from traffic counts. *Transportation Science,* 17, 198-217.

Bell, M.G.H. (1991) The estimation of origin-destination matrices by constrained generalized least squares. *Transportation Research,* 25 B, 13-22.

Chang, G.L., Tao, X. (1996) Estimation of dynamic O-Ds for urban networks. Proceedings of Thirteenth International Symposium on Transportation and Traffic Theory, Transportation and Traffic Flow Theory, 13, 1-20.

Chang, G.L., Tao, X. (1999) An integrated model for estimating time-varying network origin-destination distribution. *Transportation Research,* 33 A, 381-399.

Chang, G.L., Wu, J. (1994) Recursive estimation of time-varying origin-destination flows traffic counts in freeway corridors. *Transportation Research*, 28 B, pp.141-160.

Chiou, Y.C., Lan, L.W., Tseng, C.M. (2010) Estimation of dynamic freeway origin-destination matrices with cell-based arrival distribution modeling. *Journal of Eastern Asia Society for Transportation Studies*, 7, 1-16.

Chiou, Y.C., Lan, L.W., Tseng, C.M., Fan, C.C. (2012) Optimal locations of license plate recognition to enhance the origin-destination matrix estimation. *Asian Transport Studies,* 2, 80-92.

Chiou, Y.C., Lan, L.W., Tseng, C.M. (2013) A novel method to predict traffic features based on rolling self-structured traffic patterns. *Journal of Intelligent Transportation Systems*. (in press)

Daganzo, C.F. (1994) The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research,* 28 B, 269-287.

Hazelton, M. (2001) Inference for origin-destination matrices: estimation, prediction and reconstruction. *Transportation Research,* 35B, 667-676.

Lin, P.W., Chang, G.L. (2005) A robust model for estimating freeway dynamic origin-destination matrix. *Transportation Research Record*, 1923, 110-118.

Lin, P.W., Chang, G.L. (2007) A generalized model and solution algorithm for estimating dynamic freeway origin-destination matrix. *Transportation Research*, 41B, 554-572.

Lo, H., Zhang, N., Lam, W. (1996) Estimation of an origin-destination matrix with random link choice proportions: A statistical approach. *Transportation Research*, 30B, 309-324.

Rauber, A., Merkl, D., Dittenbach, M. (2002) The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks,* 13(6), 1331-1341.

Yang, H., Sasaki, T., Iida, Y., Akiyama, T. (1992) Estimation of origin-destination matrices from link traffic counts on congested networks. *Transportation Research,* 26B, 417-434.

Yang, H. (1995) Heuristic algorithms for the bilevel origin-destination matrix estimation problem. *Transportation Research,* 29B, 231-242.