01 Jan 2007

Visual Traffic Movement Counts at Intersection for Origin-Destination (O-D) Trip Table Estimation

S. Lee

Hojong Baik
Missouri University of Science and Technology, baikh@mst.edu

J. Park

Recommended Citation

S. Lee et al., "Visual Traffic Movement Counts at Intersection for Origin-Destination (O-D) Trip Table Estimation," *Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference (2007, Seattle, WA)*, pp. 1108-1113, Institute of Electrical and Electronics Engineers (IEEE), Jan 2007.
The definitive version is available at https://doi.org/10.1109/ITSC.2007.4357733

Proceedings of the 2007 IEEE
Intelligent Transportation Systems Conference
Seattle, WA, USA, Sept. 30 - Oct. 3, 2007

Visual Traffic Movement Counts at Intersection and Origin-Destination (O-D) Trip Table Estimation

Sang-Mook Lee, Hojong Baik, and Jae H. Park

Abstract—**Origin-destination (O-D) trip table is necessary information for transportation planning and traffic impact study. However, current O-D estimations rely on estimated directional traffic counts at intersections, which obviously diminish reliability in the table. A video-based vehicle tracking system that utilizes wide view-angle lenses has been studied [26] to provide accurate direction traffic counts. The system expands the area covered by a single camera and uses a particle filtering method to handle environmental changes as well as geometric distortion caused by those lenses. This paper shows tracking capability of the system and also shows how to incorporate the directional traffic counts in the O-D estimation.**

I. INTRODUCTION

Turning Movement Counts (TMC) at intersection is essential information in traffic operations and transportation planning [1,25]. For example, in order to design a more efficient signal timing plan for signalized intersections, it is vital to have accurate TMC on either on-line or off-line based. TMC information is also crucial in estimating the Origin-Destination (O-D) trip table [22,23] that provides information on the number of trips between zone pairs. The table is basic information for traffic impact study and transportation planning.

Currently, TMC is obtained either manually by surveyors or by traffic detectors that are installed and operated as a part of traffic control systems. Among others, loop detectors and image processors are two typical types of detection system that are mainly implemented in traffic signal control systems at urban intersections. Loop detectors are reported to operate very reliably under almost all weather condition whereas the accuracy of image processors is influenced by environmental factors such as weather, sunshine, position of the cameras, etc. Nonetheless, lower maintenance cost and flexibility in the operational features are practical reasons why image processors become more attractive device than loop detectors.

It is important to point out that the two types of detectors are not designed to obtain TMC. Instead, the devices are designed and operated to count the number of vehicles that pass over detecting locations. It is also important to realize that in many occasions, the *locational traffic counts* do not automatically provide TMC which is *directional traffic counts*. This situation happens especially when a lane is shared by two traffic flows. For example, if a lane is shared by both left-turning and through vehicles (or a lane shared by both right-turning and through vehicles), then it is impossible to get *directional traffic counts* by means of *locational traffic counts* at upstream point(s) of the intersection.

To overcome this difficulty, several modeling approaches have been proposed. Sunkarai *et al*. [1] have proposed an analytical model that computes TMC using traffic signal plans as well as traffic counts from detectors. To increase accuracy of the model, extra inductive loops at downstream are required to be installed. The accuracy of the model being computed from differences between the observed and TMCs is reported to be about 90%.

The main purposes of this paper are twofold. First, we explain how to obtain TMC using an image processing system with wide view-angle lenses. The system traces each vehicle's movement instead of just counting the number of vehicles passing certain points. It is also distinctive to apply wide view-angle lenses and to expand the area covered by a single camera, which potentially will reduce the number of cameras required to cover a whole intersection. Secondly, we describe how to use TMC information in obtaining more accurate O-D trip table.

In the next, the proposed vehicle tracking system and its tracking algorithm are described in section II and III, respectively. Then, section IV describes some background information on destination prediction. Section V shows experimental results on tracking and O-D estimation. Section VI concludes the paper.

II. VEHICLE TRACKING SYSTEM

Vehicle tracking for intelligent transportation system has been an active research area for decades and achieved some promising results especially in freeway applications [2]-[4]. However, only a few researches have been performed for both tracking and determining directional information at intersections [5,6], despite that such information is critical in constructing an O-D trip table.

Meanwhile, in order to provide the O-D estimation with exact turning movement counts, a new vehicle tracking system has been proposed [26]. The system utilizes a fisheye lens that has an extremely wide angle of view and takes in a hemispherical image. When installed properly above a center of intersection, the lens can view scenes from all directions and thus contribute to reduce deployment cost by

S-M Lee is with Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24060 USA (e-mail: lsmook@vt.edu).

H. Baik is with Department of Civil Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24060 USA (e-mail: hbaik@vt.edu).

J. H. Park is with Adaptive Genomics Corporation, Blacksburg, VA, 24060 USA (e-mail: jae.park@adaptivegenomics.com).

1-4244-1396-6/07/$25.00 ©2007 IEEE.

using fewer cameras per intersection. A drawback of using such a wide view-angle lens is a large amount of distortion. Though a geometry conversion can correct the distortion, it is not trivial to obtain conversion equation from arbitrary camera mounting angles and is desired to reduce computational burden in systems that require a real-time performance. The geometry conversion, therefore, is dropped in the system and instead the tracking performs directly on the fisheye geometry with the aids of motion dynamics models that implicitly deal with the geometric distortion.

A. Camera Installation

Current vehicle tracking systems, deployed at intersections [7,8], view only either one of upstream or downstream traffic. Moreover, their installation is affected by vertical and lateral viewing angles, number of lanes observed, and image quality, and therefore, it is required to use at least more than one camera to cover an entire intersection. However, a wide view-angle lenses, a fisheye lens and a μ-video lens as shown in Fig. 1, have a large field of view and can cover all lanes from all direction with only one camera. In most cases, the camera can be mounted on one corner of sidewalk to achieve a clear view from all direction.

B. Traffic Flow Models

Vehicle's traveling paths on image plane are fixed when a system is installed. In addition, lane types (shared or non-shared) are permanently assigned at design time. Vehicles on the non-shared lanes need not to be tracked while vehicles on shared lanes vehicles have to be tracked to determine which way they are heading. Consequently, it is necessary to assign each path of a shared lane with a motion dynamics that simulates vehicle's moving trajectory. Hence, the model parameters are highly dependent both on geometry of intersection and on camera installation. The dynamics can be trained at deployment stage. A few possible traffic flows are depicted in Fig. 2 for a 4-way intersection.

III. Tracking with Particle Filtering

A particle filtering method has been popular since a pioneering work of several researchers [9]-[11]. As one of sequential Monte Carlo frameworks, the particle filtering approximates a probability distribution of the object state by a finite set of particles, and it effectively handles non-Gaussian/nonlinear problems. The particle filtering has proved to be successful in tracking applications [12]-[14]. To apply the particle filter to vehicle tracking at intersections, two functionalities is added to the filter: a reference update scheme and a motion dynamics selection. The reference update scheme is designed to deal with fluctuations in environmental conditions such as shadows, reflection, and partial occlusion as well as variations on object's own poses. Dynamics selection is to determine direction of vehicle in shared lanes.

(a) (b) (c)

Fig. 1. Lens and a samples image. (a) Nikon FC-E8 fisheye converter (183° view-angle). (b) A μ-video lens (150° view-angle). (c) A sample image captured with the μ-video lens.

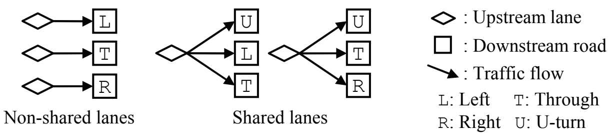


Fig. 2. Traffic flow models (TFM).

A. Bayesian Tracking and Particle Filtering

Bayesian perspective to the tracking problem is toconstruct a probability density function, $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, so that the objective tracking quantity can be determined by an expectation, $E[\mathbf{f}(\mathbf{x}_k|\mathbf{y}_{1:k})]$, of a given function, $\mathbf{f}(\mathbf{x}_k|\mathbf{y}_{1:k})$. The expression $\mathbf{y}_{1:k}$ represents all history of measurements. Then, assuming that $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ at time k–1 is known, the probability density function (PDF) $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ can be obtained by Bayes' rule:

$$\begin{aligned} p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) &= \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1} \\ p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) &= \frac{p(\mathbf{y}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1})} \end{aligned}, \quad (1)$$

where $p(\mathbf{y}_k|\mathbf{y}_{1:k-1})$ is normalization factor.

In general, the solution of this optimal filter cannot be obtained analytically, and many approximation techniques including extended Kalman filters and particle filters are developed. In particular, particle filtering outperforms (extended) Kalman filter approaches when the required PDF remains non-Gaussian throughout the iterations. If the PDF is Gaussian, Kalman filter performs satisfactory, but, in many visual tracking problems where the Gaussian assumption often breaks.

The main concept of the particle filtering is to approximate the PDF $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ with a set of particles, $\{\mathbf{x}_{k-1}^m\}_{m=1}^M$. Then, the construction of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ consists of following three steps: (1) to predict $\mathbf{x}_k^m$ at k, (2) to associate each particle with an importance weights w_k^m, and (3) to resample the particles. Here, it is commonly chosen that the importance weights are proportional to the likelihood distribution, i.e. $w_k^m \sim p(\mathbf{y}_k \mid \mathbf{x}_k^m)$. Then, the paired samples $\{(\mathbf{x}_k^m, w_k^m)\}_{m=1}^M$ properly approximate the desired PDF when M is large enough. The procedure is known as sampling importance resampling (SIR) and is popularly used for tracking problems [15].

B. Prior Distribution and Transition Model

In order to initiate the particle filtering, it is required to provide a known prior distribution $p(\mathbf{x}_0|\mathbf{y}_0) \equiv p(\mathbf{x}_0)$ from which a set of particles at k=0 are drawn. Another prior knowledge that must be provided is a state transition model which is used to predict the next state. Here, defining the state at time k as $\mathbf{x}_k = [\mathbf{d}_k, \mathbf{s}_k]$, where $\mathbf{d}_k=[x_k, y_k]$ and $\mathbf{s}_k=[s_{xk}, s_{yk}]$ are a location and a scale factor of an object being tracked in an image frame, respectively, we use a second-order auto-regression model as motion dynamics of vehicles:

$$\mathbf{D}: \mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{x}_{k-2} + \mathbf{C}\mathbf{v}_k. \quad (2)$$

The dynamics $\mathbf{D}$ is trained with a set of known trajectories of traffic flows. Here, Gaussian noise model is assumed for $\mathbf{v}_k$.

The distribution $p(\mathbf{x}_0)$ and dynamics $\mathbf{D}$ are dependent on the camera installation and the traffic flow models described in the previous section. As an example, for shared lanes in Fig. 2, $p(\mathbf{x}_0)$ could be best modeled as a bimodal distribution, and the state transition uses two different motion dynamics at early stage of tracking. However, after a few steps of tracking, the particle filter will automatically assign high weights to particles predicted from only one dynamics model and then exclude the other one for the resampling step. Fig. 3 shows an example of model selection capability of the particle filter as tracking a vehicle on a shared lane.

Fig. 3. An example of model selection. The circle shows particles with a wrong dynamics.

C. Observation Model and Importance Weight

While many other choices are possible, a normalized color histogram in hue-saturation-value color space [13] is used for object model: $h_r(n)$ for a reference object and $h_k(n)$ for a measurement associated with a particle $\mathbf{x}_k$. Then, a similarity between the two histograms is determined by Bhattacharyya distance defined as

$$B[h_r, h_k] = \left[1 - \sum_{n=1}^{N} \sqrt{h_r(n) h_k(n)}\right]^{1/2}. \quad (3)$$

This distance has a value of zero when two histograms perfectly match. Assuming Gaussian, the likelihood distribution has a relation:

$$p(\mathbf{y}_k \mid \mathbf{x}_k) \propto \exp\{-B^2[h_r, h_k]/2\sigma^2\}, \quad (4)$$

where the variance σ^2 is determined experimentally. Thus, as importance weights are chosen to be proportional to the likelihood distribution, the weight set $\{w_k^m\}_{m=1}^M$ is obtained by evaluating and normalizing the data likelihood for each particle at time k. As a result, the current location of an object being tracked is estimated as

$$\mathbf{x}_k^{est} = \mathrm{E}[\mathbf{x}_k \mid \mathbf{y}_k] \approx \sum_{m=1}^{M} w_k^m \mathbf{x}_k^m. \quad (5)$$

If the $\mathbf{x}_k^{est}$ falls in a pre-defined release zone, the tracking stops for the object.

D. Reference Update

Color model of objects being tracked is subject to vary as the objects follow their trajectories. In the fisheye lens system, lens geometry also contributes to the model variation, and, therefore, the reference model needs to be updated as time elapses. A progressive update scheme defined as

$$h_{r\_new} = \alpha h_r + (1-\alpha) h_{k'} \quad (6)$$

progressively modifies the reference model by adding a measurement ($h_{k'}$) with the highest confidence. The model update takes place at time k only if the selected particles are confident enough, in other word, when the total confidence, $\tilde{\Omega}_k = \sum_m w_k^m$, is greater than a given threshold.

IV. O-D TRIP TABLE ESTIMATION

Since early 1970s, many researchers proposed different types of approaches for estimating O-D tables using traffic count information. Recently, Sherali *et al.* [16] summarize the existing approaches with two broad categories: parameter calibration techniques and matrix estimation methods. Matrix estimation methods are further classified into statistical estimation methods and mathematical programming approaches. Most of statistical estimation methods adopt Bayesian inference techniques or least squares estimation models. On the other hand, the mathematical programming approaches seek the most-likely O-D trip table by solving mathematical programming formulations maximizing the entropy or using minimum information. An important assumption that the mathematical programming approach employs is the 'Wardrop's traffic equilibrium principle [17]'. Researches developed based on the principle include [18]-[21].

Bell et al. [22] propose a stochastic user equilibrium path flow estimator (PFE) to estimate an O-D table using locational traffic count. The basic formulation is given by:

$$\text{Minimize} \quad \sum_{ij \in IJ} \sum_{k \in K_{ij}} f_k^{ij} (\ln f_k^{ij} - 1) + \theta \sum_a \int t_a(v_a) dv_a \quad (7)$$

$$\text{Subject to:} \quad x_a = \sum_{ij} \sum_k f_k^{ij} \delta_{ka}^{ij}, \ \forall a \quad (8)$$

$$v_a = x_a, \forall a \in M \quad (9)$$

$$x_a \le C_a, \ \forall a \in U. \quad (10)$$

where,

M,U: sets of measured and unmeasured links,

IJ and K_{ij}: sets of O-D pairs and paths connecting origin i and destination j,

θ: sensitivity of the assignment to path cost,

v_a, x_a: Observed and estimated flows on link a,

C_a, $t_a(\cdot)$: Capacity and cost function of link a,

f_k^{ij} : estimated flow on path k connecting origin i and destination j,

δ_{ak}^{ij} : path-link indicator, 1 if link a paths k between origin i and destination j, 0 otherwise.

The objective function (7) minimizes the sum of an entropy term and the total travel cost term. Equation (8) is to convert the estimated path flow to the link flow. Constraint

(9) makes estimated traffics for measured links equal to observed traffic counts on the same links. Constraint (10) limits flows on the unobserved link to be under the link capacity. Chen et al. [23] extend the model by adding constraints to control the estimation error defined as the difference between estimated and observed link flows to be within a certain error interval.

As a solution process, the Lagrangian equation of the original formulation is obtained by introducing Lagrangian multipliers (i.e., dual variables) for constraints (9) and (10). Path flows together with values of Lagrangian multipliers are obtained applying the iterative balancing method proposed by Bell and Shield [22]. The method considers each constraint at a time by adjusting the corresponding dual variable. (See [22] for more details.)

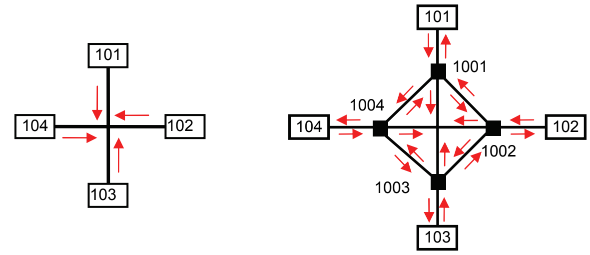
It should be noted that the formulation (7)-(10) is based on link flows obtained from the locational traffic counts. In order for the model to use TMC information from the tracking system described in this paper, the topology of the road network needs to be adjusted. Fig. 4 shows the new structure of an intersection. The path for left-turning vehicles from the node 103, for example, is 103→1003→1004→104. The total link flow observed from TMC for the same movement is assigned on the link of 1003→1004.

Another adjustment is to add terminal nodes that represent drivers' final destination or trip origin points (for example parking lots of buildings). The nodes are connected to road links with additional connectors. This adjustment is made to avoid the violation of the traffic flow conservation rule between intersections. Fig. 5 illustrates the adjusted network with terminal nodes (9, 10, 11 and 12) connected to additional nodes (17 and 22) with dotted lines. The adjusted network involves total 12 nodes, which indicates that the O-D trip table needs to be a matrix with the size of 12x12. Now, the mathematical formulation given in (7)-(12) can be applied to the adjusted network using TMC measured from the tracking system.

V. EXPERIMENTAL RESULTS

A. *Vehicle Tracking*

Two video sequences are used for the feasibility test of the proposed acquisition system and tracking algorithm. One with toy cars, taken with the FC-E8 fisheye converter, simulates an ideal installment of the camera, right above the center of intersection providing a symmetric scene with the same geometric distortion for all directions. The second sequence is recoded at the rate of 30 frames per second with the micro video wide-angle lens mounted on a pole of 4.3 meter high. The pole was installed at a corner of intersection and the distance to the opposite corner is about 30 meters. The tracking algorithm is implemented with Matlab™ and processes 2 to 4 frames per second on a Pentium 4 2.8GHz IBM compatible computer. Parameters are set to σ^2=0.05, $\tilde{\Omega}_k$ =0.6M and α=0.9 throughout the experiments.



(a) Without TMC (b) With TMC

Fig. 4. Intersection diagrams.

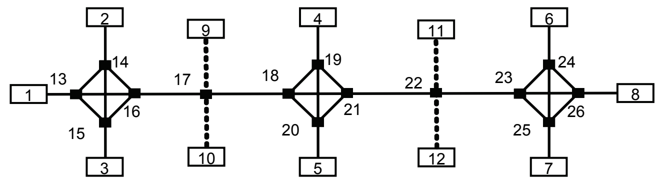
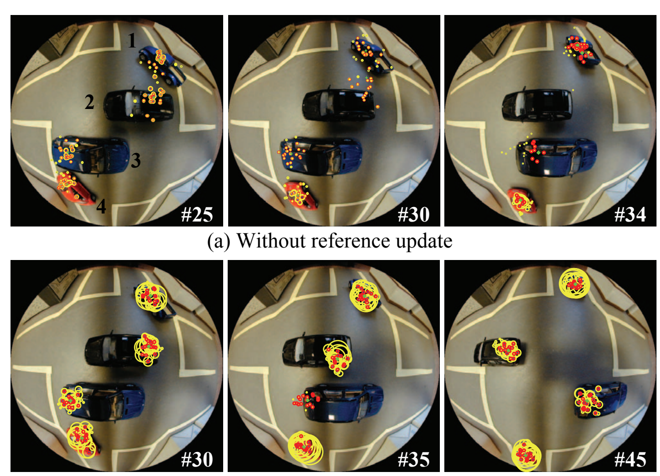


Fig. 5. The adjusted network.

First, the Fig. 6 shows tracking results of toy cars. Notice that there are large change of scale and pose and also there exist vertical shadows around the center of frames. Under these poor conditions, the reference update scheme is critical for the filter to correctly track the objects. A tracking without the reference update easily looses objects mainly due to shadow cast, reflection or varying poses (Fig. 6a). The particles have a considerably low confidence level at the frame 25 for all vehicles. At frame 30, most of particles for car 2 started drifting to car 1 and then later all the particles assigned to the car 2 follow the car 1. However, the car 4 has low confidence levels at frame 25 but recovers at 34 because the different pose has the similar look to its first pose. On the other hand, Fig. 6b shows an effect on the reference update scheme. All the particles throughout the sequence have high confidence levels except the car 3 at frame 35. The size of yellow circles corresponds to the confidence level.

In the second sequence, vehicles entering the intersection are manually selected since a windy weather condition caused an ego-motion of camera and thus many undesired noise in applying a background subtraction technique.



(a) Without reference update

(b) With reference update

Fig. 6. A center mount and effect on reference update scheme. M = 20.

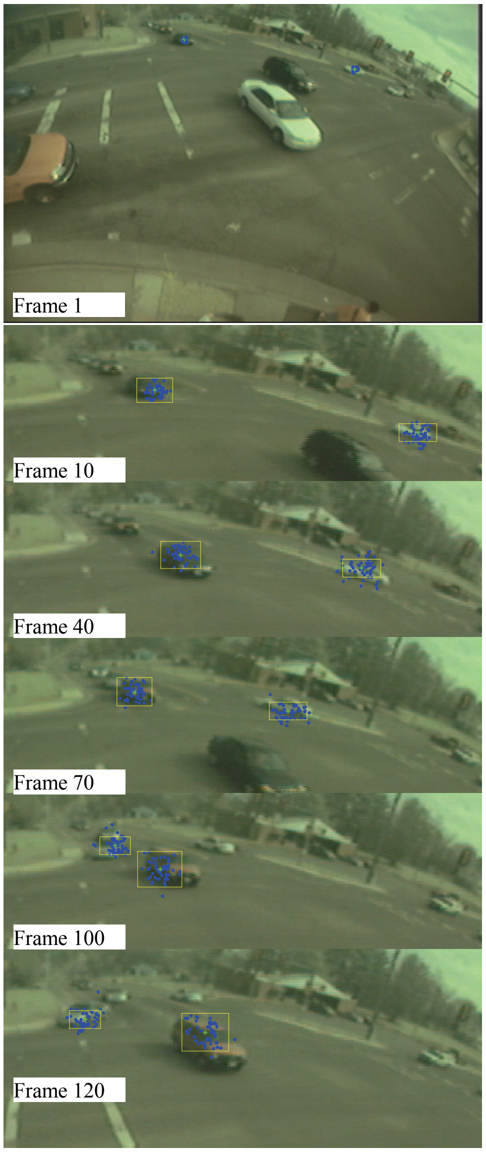


Fig. 7. Corner mount and tracking at real situation. M=50.

For the background subtraction to be applicable in this situation, a scene registration must be done a prior, which requires more computing power and is left for further study. A tracking result for the sequence is shown in Fig. 7. Image formant is 720×480, and 50 particles are used and their sizes vary from 12×24 to 35×50. The vehicles are tracked until they reach pre-defined release zones at which vehicle directions are surely determined. Here, blue dots indicate locations of particles and the green stars represent estimated vehicle positions.

B. O-D Trip Table Estimation

O-D estimation procedure is conducted using TMC information along with roadway design characteristics. Table 1 shows link information hypothetically generated for the adjusted network illustrated in Fig 5. Here, capacities of links connecting terminal nodes and road links (for example, link 9→17) represent maximum flow rates that can be defined by a combination of total number of entrances to parking lots in the street block and the average time for a vehicle to enter or exit the parking lots.

It is assumed that there are two types of links: 1) links where TMC is measured using the tracking system proposed in this paper, and 2) links where no measurement is made. In the adjusted network, all links consisting of intersections are assumed to be measured by the tracking system, and links connecting terminal nodes and road links not measured. The last column in Table 1 shows the traffic counts. (This assumption is made for the illustration purpose. The PFE method can be applied to any combination of link sets.)

The solution algorithm proposed by Bell et al. [22] is applied to obtain an O-D table that satisfies the Karush-Kuhn-Tucker (KKT) condition being the necessary and sufficient condition for the optimality of the Lagrangian equation. For the computational simplicity, a volume-delay function proposed by the Bureau of Public Road [24] is used. Table 2 shows the resulting O-D table using the information given in Table 1. Numbers of cars traveled from and to the terminal node 9 are 80 and 126 vehicles respectively which are exactly same as ones traveled from and to the terminal node 10.

TABLE 1. LINK CHARACTERISTICS OF THE ADJUSTED NETWORK IN FIG. 5.

Link	From	To	Capa (vph)	Speed (mph)	Dist (mile)	Count (vph)	Link	From	To	Capa (vph)	Speed (mph)	Dist (mile)	Count (vph)
1	1	13	3600	35	0.1313	2428	35	20	21	1000	5	0.006	138
2	13	1	3600	35	0.1313	2937	36	22	21	3600	35	0.0618	2441
3	13	14	1000	5	0.019	46	37	21	22	3600	35	0.0618	2326
4	13	16	3600	35	0.015	2225	38	21	20	1000	5	0.019	182
5	13	15	1000	5	0.006	157	39	21	18	3600	35	0.015	2075
6	2	14	1800	25	0.1347	277	40	21	19	1000	5	0.006	184
7	14	2	1800	25	0.1347	256	41	22	23	3600	35	0.0618	2415
8	14	16	1000	5	0.019	30	42	23	22	3600	35	0.0618	2236
9	14	15	1800	25	0.015	142	43	23	24	1000	5	0.019	64
10	14	13	1000	5	0.006	105	44	23	26	3600	35	0.015	2025
11	3	15	1800	25	0.1313	613	45	23	25	1000	5	0.006	326
12	15	3	1800	25	0.1313	356	46	6	24	1800	25	0.1347	695
13	15	13	1000	5	0.019	411	47	24	6	1800	25	0.1347	824
14	15	14	1800	25	0.015	112	48	24	26	1000	5	0.019	126
15	15	16	1000	5	0.006	90	49	24	25	1800	25	0.015	521
16	17	16	3600	35	0.0618	2576	50	24	23	1000	5	0.006	48
17	16	17	3600	35	0.0618	2345	51	7	25	1800	25	0.1313	1146
18	16	15	1000	5	0.019	57	52	25	7	1800	25	0.1313	1167
19	16	13	3600	35	0.015	2421	53	25	23	1000	5	0.019	354
20	16	14	1000	5	0.006	98	54	25	24	1800	25	0.015	626
21	17	18	3600	35	0.0618	2431	55	25	26	1000	5	0.006	166
22	18	17	3600	35	0.0618	2570	56	8	26	3600	35	0.1313	2288
23	18	19	1000	5	0.019	109	57	26	8	3600	35	0.1313	2317
24	18	21	3600	35	0.015	2026	58	26	25	1000	5	0.019	320
25	18	20	1000	5	0.006	296	59	26	23	3600	35	0.015	1834
26	4	19	1800	25	0.1347	663	60	26	24	1000	5	0.006	134
27	19	4	1800	25	0.1347	624	61	9	17	500	5	0.006	-
28	19	21	1000	5	0.019	162	62	17	9	500	5	0.006	-
29	19	20	1800	25	0.015	375	63	10	17	300	5	0.005	-
30	19	18	1000	5	0.006	126	64	17	10	300	5	0.005	-
31	5	20	1800	25	0.1313	838	65	11	22	500	5	0.006	-
32	20	5	1800	25	0.1313	853	66	22	11	500	5	0.006	-
33	20	18	1000	5	0.019	369	67	12	22	300	5	0.005	-
34	20	19	1800	25	0.015	331	68	22	12	300	5	0.005	-

TABLE 2. ESTIMATED O-D (θ= 0.1).

From \ To	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Total**
1	0	46	157	97	264	47	237	1474	29	29	24	24	**2428**
2	105	0	142	1	4	1	3	20	0	0	0	0	**277**
3	411	112	0	4	11	2	10	60	1	1	1	1	**613**
4	114	5	3	0	375	4	21	132	2	2	2	2	**663**
5	334	14	8	331	0	4	18	113	7	7	2	2	**838**
6	36	1	1	4	4	0	521	126	1	1	0	0	**695**
7	268	11	6	26	26	626	0	166	6	6	3	3	**1146**
8	1387	56	33	136	134	134	320	0	29	29	15	15	**2288**
9	48	2	1	3	9	2	8	51	0	1	1	1	**126**
10	48	2	1	3	9	2	8	51	1	0	1	1	**126**
11	93	4	2	9	9	2	10	62	2	2	0	1	**196**
12	93	4	2	9	9	2	10	62	2	2	1	0	**196**
Total	2937	256	356	624	853	824	1167	2317	80	80	50	50	**9593**

This happens because for the computational simplicity, no left-turn prohibition is applied to traffics traveled from the terminal nodes. Under this condition, terminal nodes 9 and 10 equally contribute to compensating the flow difference on links 16↔17 and 17↔18. The resulting net traffic from nodes 9 and 10 is 92 (=252-160) vehicles which is the exact number of vehicles required to satisfy the flow conservation on the links 16↔17 and 17↔18. The links have total 4915 (=2345+2570) inflows and 5007 (=2431+2576) outflows.

The value of θ representing the sensitivity of the assignment to path cost is currently set to 0.1. It should be noted that the estimated O-D table is independent of the value of θ. This is because the example network permits only single route between O-D pairs, which makes it unnecessary for a car to consider multiple routes based on travel time. This means that in case of the single path problem, the travel time term (i.e., the second term) in the formulation just minimizes the total travel time, but does not affect resulting path flows. In case of multi path problem, calibration process is necessary to find an appropriate θ value that represents the observed travel time closely.

VI. Conclusion

The proposed system tracks vehicles on shared lanes and determines their direction, which is impossible for existing systems. In the particle filtering, the most time consuming task is the measurement that constructs color histograms. Fortunately, however, the procedure is exactly same for all particles and can be readily implemented by using a parallel computing device such as FPGA (Field Programmable Gate Array). A limitation of using color histograms as an object model is that it works only on day situations.

In order to estimate O-D table, the given road network is appropriately adjusted so that it can explicitly represent the turning movement counts, and a stochastic user equilibrium PFE is employed. It is natural that more information on traffic movements contributes to estimation of higher accurate O-D table.

Our future work will include 1) making the tracking robust so that it can handle various weather conditions, 2) implementing the part of algorithm with FPGA, and 3) developing an algorithm that updates the O-D table in real-time. Future works related to the O-D estimation are: 1) applying the PFE algorithm to a grid type of network that involves multiple paths for an O-D pair, 2) comparing two O-D tables estimated with and without using TMC information at shared lanes, and 3) considering the maximum parking space of each terminal node which requires to add another constraint in the formulation and to modify the Lagrangian equation accordingly.

References

[1] S. R. Sunkari, H. Charara and T. Urbanik. "Automated turning movement counts from shared-lane configuration at signalized diamond interchanges," *79th Annual Meeting of the Transportation Research Board*. Washington, D.C. January 2000.

[2] I. Ikeda, S. Ohnaka, and M. Mizoguchi, "Traffic measurement with a roadside vision system-individual tracking of overlapped vehicles," *Proc. Intl Conf. Pattern Recognition*, 1996, pp.859-864.

[3] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997 pp.495-501.

[4] A.N. Rajagopalan, R. Chellappa, "Vehicle detection and tracking in video," *Proc. Intl Conf. Image Processing*, vol.1, 2000, pp.351-54.

[5] C. Dalaff, R. Reulke, A. Kroen, T. Kahl, M. Ruhe, A. Schischmanow, G. Schlotzhauer, and W. Tuchscheerer, "A Traffic object detection system for road traffic measurement and management," *Proc. Image and Vision Computing New Zealand*, ed. D. Bailey, 2003, pp. 78-83.

[6] S. Chen, M. Shyu, S. Peeta, and C. Zhang, "Learning-based spatio-temporal vehicle tracking and indexing for transportation multimedia database systems," *IEEE Trans. Intelligent Transportation Systems*, vol.4, no.3, Sept., 2003, pp. 154-167.

[7] Autoscope, www.imagesensing.com

[8] Y. Anzai, T. Kato, M. Higashikubo, K. Tanaka, and T. Hinenoya, "Development of an integrated video imaging vehicle detector," *SEI Technical Review*, no.59, Jan. 2005, pp. 43-47.

[9] N. Gordon and D. Salmond, "Bayesian state estimation for tracking and guidance using the bootstrap filter," *J. of Guidance, Control and Dynamics*, vol. 18(6), 1995, pp.1434-1443.

[10] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," *European Conf. Computer Vision*, vol. 1, 1996, pp. 343-356.

[11] M. Isard and A. Blake, "CONDENSATION-Conditional density propagation for visual tracking," *Intl J. on Computer Vision*, vol. 1(29), 1998, pp. 5-28.

[12] M. Isard and J. MacCormick, "BraMBLe: a Bayesian multiple-blob tracker," *Proc. IEEE Intl Conf. Computer Vision*, vol.2, 2001, pp.34-41.

[13] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," *European Conf. Computer Vision*, 2002, pp. 661-675.

[14] J. Vermaak, A. Doucet, and P. Pérez, "Maintaining Multi-modality through mixture tracking," *Proc. IEEE Intl Conf. Computer Vision*, vol. 2, 2003, pp.1110-1116.

[15] A. Doucet, N. de Freitas, and N. Cordon, editors, *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, 2001.

[16] H.D. Sherali, A. Narayanan and R. Sivanandan, "Estimation of origin–destination trip-tables based on a partial set of traffic link", *Transportation Research*, 37B, 2003, pp. 815-836.

[17] W.A. O'Neill, *Origin-destination trip table estimation using traffic counts*. Ph.D. Dissertation, University of New York at Buffalo, NY, 1987.

[18] J.G. Wardrop, "Some Theoretical Aspects of Road Traffic Research." *Proc. Institution of Civil Engineers*, Part II-1, 1952. pp.325-378.

[19] C. Fisk and D. E. Boyce, "A note on trip matrix estimation from link traffic count data," *Transportation Research*, 17B, 1983, pp.245-250.

[20] E. Cascetta, and S. Nguyen, "A unified framework for estimating or updating origin/destination matrices from traffic counts", *Transportation Research*, 22B, 1988, pp. 437-455.

[21] H. Yang, "Heuristic algorithms for the bi-level algorithm," *Transportation Research*, 29B, 1995, pp. 231-242.

[22] M.G.H. Bell, C.M. Shield, F. Busch and G. Kruse, "A stochastic user equilibrium path flow estimator," *Transportation Research*, 5C, 1997, pp. 197-210.

[23] A. Chen, P. Chootinan, W. Recker, and H. M. Zhang, "Development of a Path Flow Estimator for Deriving Steady-State and Time-Dependent Origin-Destination Trip Tables," *Research Report (UCI-ITS-WP-o4-13)*, University of California, Berkeley. 2004.

[24] Bureau of Public Roads, *Traffic Assignment Manual*, U.S. Dept. of Commerce, Urban Planning Division, Washington D.C. 1964

[25] G. A. Davis and C-J Lan, "Estimating intersection turning movement proportions from less-than-complete sets of traffic counts", *Transportation Research Board*, 1995, pp. 53-59.

[26] S-M Lee and H. Baik, "Origin-Destination (O-D) Trip Table Estimation using Traffic Movement Counts from Vehicle Tracking System at Intersection", in *Proc. of The 32nd Annual Conf. of IEEE Industrial Electronics Society*, Nov. 2006, pp. 3332-3337.