RC25591 (WAT1602-016) February 15, 2016
Computer Science

IBM Research Report

City-Wide Traffic Flow Estimation from Limited Number of Low Quality Cameras

Tsuyoshi Idé[1], Takayuki Katsuki[2],
Tetsuro Morimura[2], Robert Morris[1]

[1]IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598 USA

[2]IBM Research - Tokyo
IBM Japan, Ltd.
19-21, Hakozaki-cho, Nihombashi, Chuoh-ku
Tokyo 103-8501, Japan

City-Wide Traffic Flow Estimation from Limited Number of Low Quality Cameras

Tsuyoshi Idé, Takayuki Katsuki, Tetsuro Morimura, and Robert Morris

***Abstract*—We propose a new approach to intelligent transportation systems for developing countries. Our system consists of two major components: (1) Web-camera-based traffic monitoring and (2) network flow estimation. The traffic monitoring module features a new algorithm for computing the vehicle count from very low-resolution webcam images. To reduce the cost of camera-wise collection of labeled (i.e. manually counted) images, we develop a novel unsupervised learning approach. The network flow estimation module features a traffic flow estimation algorithm formalized as an inverse Markov chain problem, which finds the entire flow matrix from partial observations using an information-theoretic criterion. Using real webcams deployed in Nairobi, Kenya, we demonstrate the utility of our approach.**

I. Introduction

TRAFFIC congestion is a major problem in the urban regions of most developing countries, where mismatches are found between rapidly growing economies and the municipal infrastructures. Intelligent transportation systems (ITS) provide a basic framework for traffic management. Unlike urban areas in relatively mature countries, cities with rapid economic growth require a lightweight ITS to adapt to the dynamically changing environment.

What we are interested in here is a "Frugal" approach [1] to ITS. Instead of relying heavily on expensive infrastructures such as an inductive-loop sensor system covering an entire city area for traffic monitoring, we wish to develop an ITS that is easy to deploy, has a minimum entry cost, and offers good enough functionalities. As an alternative to the existing full-scale systems, we focus on a webcam-based monitoring approach. Webcam-based traffic surveillance through Web browsers is already available in many cities in developing countries. For instance, in Nairobi City, Kenya, AccessKenya.com [2] runs a Web site to provide near real-time information on the traffic at major locations. Although just looking at the webcam images through Web browsers is useful enough for personal use, it is not the case for traffic authorities. For the purpose of city planning and traffic optimization, we need to extract key information of traffic flow from webcam images for the entire city. This is indeed the main topic of this paper.

To make camera-based traffic monitoring truly useful, a lot of research has been made to date. Examples include vehicle recognition for traffic volume estimation [3], [4], [5], [6] and regression modeling for vehicle counting [7], [8]. Also, origin-destination (OD) matrix estimation algorithms [9], [10], [11], [12] are often combined with traffic estimation methods since the number of cameras is always limited [13] (See Section IV for details of related work). Although these pieces of work made significant contributions to several individual technical issues, major challenges still remain as summarized below.

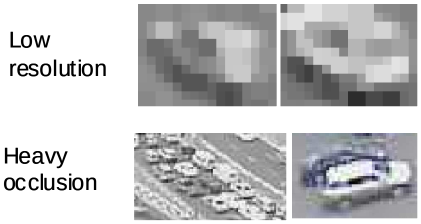

Fig. 1. Examples of very low resolution images.

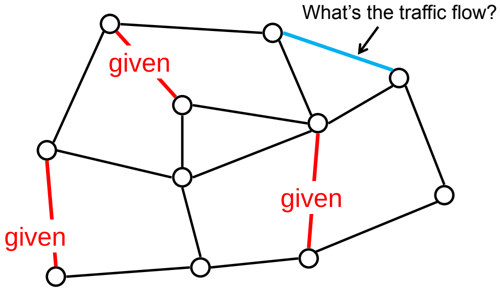

Fig. 2. Network flow inference problem.

The first challenge is how to handle very low-resolution images (see Fig. 1 for examples). Due to cost and antitheft concerns, special-purpose close-view cameras are not suitable in most developing countries. The use of general-purpose cameras without purpose-built lighting facilities makes hardly useful standard object recognition technologies such as those used in number plate recognition [6], [14].

The second challenge is how to eliminate the time-consuming step of camera-wise calibration in the image processing. Most of the recent studies on video-based ITS focus on calibration algorithms when surveillance cameras do not allow calibration on the hardware side [3], [4], [5], [6]. In either case, however, as long as vehicle recognition is performed on images, camera-wise fine adjustments based on the geometric configuration of cameras and roads are required. Although the use of regression models [7], [8] reduces the burden, a fair amount of *labeled* training data (i.e. manually counted or recognized images) is still required.

The third challenge is how to derive city-wide information from a limited number of webcams (see Fig.2). In particular, what-if simulation for optimized city planning calls for estimating the traffic volume in every single link of the road

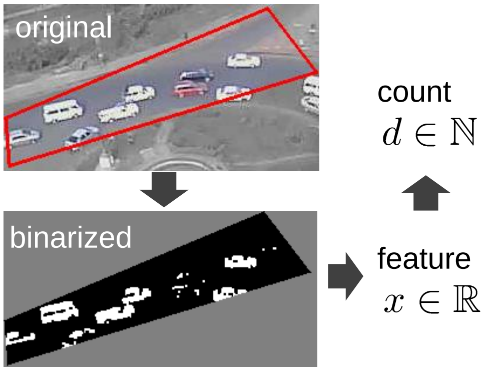

Fig. 3. Vehicle counting from low-resolution images. The symbol $\mathbb{R}$ and $\mathbb{N}$ denote the real and natural numbers (non-negative integers), respectively.

network. This task is similar to network tomography for the OD matrix [3], [4], [5], [6], but differs in that we need to infer the traffic volume at all of the links instead of just the origin-destination flows.

We tackle these challenges using novel machine learning techniques. Our technical contributions are as follows:

- We developed a novel algorithm for fully calibration-free vehicle counting. One prominent feature of our method (see Section II) is that it requires neither camera-wise calibration nor labeled data generation.
- We developed a new inference algorithm on road networks to estimate the traffic volume at arbitrary links without direct observation by webcams. We formalize the problem as an inverse Markov chain problem, and leverage a mathematical technique of regularization (see Section III).

These methods are validated with real webcam traffic images in Nairobi, Kenya, as elaborated in Section V.

These methods are already outlined in a preliminary version of this paper [15]. This paper significantly expands it by adding algorithmic and experimental details, based on our companion papers [16], [17] that focus more on theoretical aspects of the vehicle counting and the network inference problems, respectively.

II. Calibration-free vehicle counting

This section presents an approach to calibration-free low-resolution image analysis for vehicle counting. The approach consists of two major steps (Fig. 3). The first step is to extract a feature value from raw images, as discussed in Section II-A. The second step is to estimate the count of vehicles contained in the region of interest of images, as discussed in Section II-B and on. Since the webcams are analyzed independently, we focus on a single webcam in the rest of this section.

A. Feature extraction

Let N be the number of training images for one camera we are focusing on. Assume that all the images have the same M pixels, and each of the pixels takes an integer from the 256 luminance levels. Our data set is represented as

$$\mathcal{D} = \{\boldsymbol{z}^{(n)} \in \{0, 1, 2, \dots, 255\}^M \mid n = 1, \dots, N\}. \tag{1}$$

For each image, as preprocess, we subtract the median over the M pixels to normalize the variation of overall luminance. This is useful to handle the variation e.g. between nighttime and daytime. The goal of feature selection is to extract a feature $x \in \mathbb{R}$ from a raw image $\boldsymbol{z} \in \{0, \dots, 255\}^M$ such that x corresponds to a rough estimate of the count of vehicles.

As indicated in Fig. 3, our approach first convert the original image into a binarized image. Once binarization is done, the feature x is computed as the ratio of white pixels to the total number of pixels:

$$x = \frac{1}{M} \sum_{i=1}^{M} I(z_i \geq k^*), \tag{2}$$

where k^* is the threshold for binarization to be determined from the data, and $I(\cdot)$ is the indicator function that gives 1 when the argument is true, and 0 otherwise.

Figure 4 shows an example of the distribution of pixel values of the training set, where the vertical bar separates the black (called class 1) from the white (class 2) pixels. Let k be the value of the threshold. To find the optimal value $k = k^*$, we follow Otsu's method [18]. The idea is to maximize the statistical variance between the two classes. If we think of Fig. 4 as a probability distribution, the probability of the l-th luminance level is given by

$$p_l \equiv \frac{1}{MN} \sum_{n=1}^{N} \sum_{i=1}^{M} I(z_i^{(n)} = l). \tag{3}$$

Based on this, the total mean luminance is given by

$$\bar{\ell} \equiv \sum_{l=0}^{255} p_l l. \tag{4}$$

Similarly, the mean luminance for the class 1 and class 2 is given by

$$\ell_1(k) \equiv \frac{1}{P_1(k)} \sum_{l=0}^{k-1} p_l l, \quad \ell_2(k) \equiv \frac{1}{P_2(k)} \sum_{l=k}^{255} p_l l, \tag{5}$$

where $P_1(k) \equiv \sum_{l=0}^{k-1} p_l$ and $P_2(k) \equiv \sum_{l=k}^{255} p_l$. Obviously, these are functions of the threshold k. Now the optimal threshold k^* is determined by solving the following optimization problem

$$k^* = \arg\max_k \left\{ \left[\ell_1(k) - \bar{\ell}\right]^2 P_1(k) + \left[\ell_2(k) - \bar{\ell}\right]^2 P_2(k) \right\}. \tag{6}$$

This can be simply solved by evaluating the objective function for all the 256 different values, and pick one giving the maximum. Once k^* is obtained, Eq. (2) gives the feature for any image taken with the camera of interest.

B. Probabilistic counting framework

Given the optimized threshold k^*, the training data $\mathcal{D}$ is now converted into

$$\mathcal{D}' = \{x^{(n)} \in \mathbb{R} \mid n = 1, \dots, N\}. \tag{7}$$

In practice, it is recommended to further standardize the feature as $x^{(n)} \leftarrow x^{(n)} / \max_{n'} x^{(n')}$ before model fitting. As

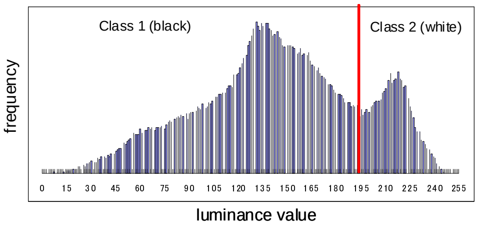

Fig. 4. Example of Luminance distribution.

discussed in Introduction, in the city-wide traffic monitoring scenario using low-resolution cameras, collecting labeled images (i.e. manually counted images) for each of the cameras is quite costly. Here we propose fully *unsupervised* approach to vehicle counting.

The vehicle counting part in Fig. 3 consists of two substeps. *First*, we find the predictive distribution for the feature, x, in the form of an infinite Gaussian mixture model

$$p(x \mid \mathcal{D}') = \sum_{d=0}^{\infty} \pi_d(x)\, \mathcal{N}\left(x \,\middle|\, \boldsymbol{m}^\top \boldsymbol{\phi}_d, \sigma_d^2\right), \tag{8}$$

where d denotes the number of vehicles, $^\top$ is the transpose, and

$$\boldsymbol{\phi}_d \equiv \begin{pmatrix} 1 \\ d \end{pmatrix}. \tag{9}$$

The parameters $\pi_d, \boldsymbol{m}, \sigma_d^2$ are learned from the data, as explained later. For the explicit definition of the Gaussian distribution $\mathcal{N}(\cdot \mid \cdot, \cdot)$, see Appendix. *Second*, we associate an observation $x = x'$ with one mixture component via

$$d' = \arg\max_d \left\{ \pi_d(x')\, \mathcal{N}\left(x' \,\middle|\, \boldsymbol{m}^\top \boldsymbol{\phi}_d, \sigma_d^2\right) \right\}. \tag{10}$$

The next section explains how to find the distribution (8).

C. Observation model and prior distributions

Although the vehicle counting approach outlined above may look like a simple density estimation problem, there are a couple of challenges we have to tackle. The first challenge is to handle the interchangeability of cluster labels. The second challenge is to handle the unbounded nature of the count.

To address these two challenges, we introduce a Bayesian density estimation model [16]. We first define the observation process by

$$p(x \mid \boldsymbol{h}, \boldsymbol{\theta}, \lambda) \equiv \prod_{d=0}^{\infty} \mathcal{N}(x \mid \boldsymbol{\theta}^\top \boldsymbol{\phi}_d, \lambda^{-1})^{h_d}, \tag{11}$$

where $\boldsymbol{h}$ is an infinite dimensional indicator vector all of whose entries are 0 except for only one entry that is 1. The key idea is to suppress the interchangeability in different d's by setting the following form of prior distribution for $\boldsymbol{h}$:

$$p(\boldsymbol{h} \mid \boldsymbol{v}) \equiv \prod_{d=0}^{\infty} \left\{ v_d \prod_{k=0}^{d-1} (1 - v_k) \right\}^{h_d}, \tag{12}$$

$$p(\boldsymbol{v}) \equiv \prod_{d=0}^{\infty} \mathrm{Beta}(v_d \mid 1, \beta), \tag{13}$$

where β is a hyper-parameter treated as a given constant and Beta is the beta distribution (see Appendix). These distributions are commonly called the stick-breaking process (SBP). As clearly indicated in the definition, the SBP prior is not symmetric in the cluster index d, and naturally introduces the order in the components.

For the other parameters $\boldsymbol{\theta}, \lambda$, we set conjugate priors as

$$p(\boldsymbol{\theta} \mid \boldsymbol{m}_0, \Sigma_0) \equiv \mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{m}_0, \Sigma_0), \tag{14}$$

$$p(\lambda \mid a_0, b_0) \equiv \mathrm{Gam}(\lambda \mid a_0, b_0), \tag{15}$$

where $\boldsymbol{m}_0, \Sigma_0, a_0, b_0$ are hyper-parameters treated as given constants, and Gam is the gamma distribution (see Appendix).

D. Variational Bayes solution

To derive the predictive distribution Eq. (8), we need to find the posterior distributions for $\boldsymbol{h}, \boldsymbol{\theta}, \lambda, \boldsymbol{v}$. This can be systematically done via the variational Bayes (VB) algorithm [19]. The VB approach approximately finds the posterior distribution in a factorized form:

$$p_{\mathrm{post.}}(\mathsf{H}, \boldsymbol{\theta}, \lambda, \boldsymbol{v}) = q(\mathsf{H}) q(\boldsymbol{\theta}) q(\lambda) q(\boldsymbol{v}), \tag{16}$$

where we used the same symbol q to represent different distributions for simplicity of notation.

The VB algorithm starts with writing down the complete likelihood as

$$P(\mathcal{D}', \mathsf{H}, \boldsymbol{\theta}, \lambda, \boldsymbol{v}) \equiv \prod_{n=1}^{N} p(x^{(n)} \mid \boldsymbol{h}^{(n)}, \boldsymbol{\theta}, \lambda) p(\boldsymbol{h}^{(n)} \mid \boldsymbol{v}) \times p(\boldsymbol{\theta} \mid \boldsymbol{m}_0, \Sigma_0) p(\lambda \mid a_0, b_0) p(\boldsymbol{v}), \tag{17}$$

where H represents $\{\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(N)}\}$. The main result of the VB algorithm is that the posterior distributions are given by the following simultaneous equations:

$$\ln q(\mathsf{H}) = \text{const.} + \langle \ln P(\mathcal{D}', \mathsf{H}, \boldsymbol{\theta}, \lambda, \boldsymbol{v}) \rangle_{\boldsymbol{\theta}, \lambda, \boldsymbol{v}}, \tag{18}$$

$$\ln q(\boldsymbol{\theta}) = \text{const.} + \langle \ln P(\mathcal{D}', \mathsf{H}, \boldsymbol{\theta}, \lambda, \boldsymbol{v}) \rangle_{\mathsf{H}, \lambda, \boldsymbol{v}}, \tag{19}$$

$$\ln q(\lambda) = \text{const.} + \langle \ln P(\mathcal{D}', \mathsf{H}, \boldsymbol{\theta}, \lambda, \boldsymbol{v}) \rangle_{\mathsf{H}, \boldsymbol{\theta}, \boldsymbol{v}}, \tag{20}$$

$$\ln q(\boldsymbol{v}) = \text{const.} + \langle \ln P(\mathcal{D}', \mathsf{H}, \boldsymbol{\theta}, \lambda, \boldsymbol{v}) \rangle_{\mathsf{H}, \boldsymbol{\theta}, \lambda}, \tag{21}$$

where $\langle \cdot \rangle_*$ represents the expectation w.r.t. the random variables $*$. By simply expanding the $\ln P$ term, we can easily see that the posterior distributions take the following forms:

$$q(\mathsf{H}) = \prod_{n=1}^{N} \prod_{d=0}^{\infty} \left\{ \pi_d^{(n)} \right\}^{h_d^{(n)}}, \tag{22}$$

$$q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{m}, \Sigma), \tag{23}$$

$$q(\lambda) = \mathrm{Gam}(\lambda \mid a, b), \tag{24}$$

$$q(\boldsymbol{v}) = \prod_{d=0}^{\infty} \mathrm{Beta}(v_d \mid \alpha_d, \beta_d), \tag{25}$$

where $\pi_d^{(n)}, \boldsymbol{m}, \Sigma, a, b, \alpha_d, \beta_d$ are unknown parameters to be determined.

To find these parameters, first we assume that $q(\mathsf{H})$ is given. Using well-known properties of the Gaussian, gamma, and beta distributions such as $\langle \boldsymbol{\theta} \rangle_{\boldsymbol{\theta}} = \boldsymbol{m}$ and $\langle \lambda \rangle_{\lambda} = \frac{a}{b}$, we easily see that the parameters satisfy the following relations:

$$N_d = \sum_{n=1}^{N} \pi_d^{(n)}, \quad \bar{x}_d = \sum_{n=1}^{N} \pi_d^{(n)} x^{(n)}, \tag{26}$$

$$\Delta_d(x^{(n)}) = (x^{(n)} - \boldsymbol{\phi}_d^\top \boldsymbol{m})^2 + \boldsymbol{\phi}_d^\top \mathsf{\Sigma} \boldsymbol{\phi}_d, \tag{27}$$

$$a = a_0 + \frac{N}{2}, \tag{28}$$

$$b = b_0 + \frac{1}{2} \sum_{n=1}^{N} \sum_{d=0}^{\infty} \pi_d^{(n)} \Delta_d(x^{(n)}), \tag{29}$$

$$\mathsf{\Sigma} = [\mathsf{\Sigma}_0^{-1} + \frac{a}{b} \sum_{d=0}^{\infty} N_d \boldsymbol{\phi}_d \boldsymbol{\phi}_d^\top]^{-1}, \tag{30}$$

$$\boldsymbol{m} = \mathsf{\Sigma}^{-1} [\mathsf{\Sigma}_0^{-1} \boldsymbol{m}_0 + \frac{a}{b} \sum_{d=0}^{\infty} \bar{x}_d \boldsymbol{\phi}_d], \tag{31}$$

$$\alpha_d = 1 + N_d, \quad \beta_d = \beta + \sum_{k=d+1}^{\infty} N_k. \tag{32}$$

To compute $\{\pi_d^{(n)}\}$, we assume in turn that $q(\boldsymbol{\theta}), q(\lambda)$ and $q(\boldsymbol{v})$ are given. Expanding the $\ln P$ term and taking expectation w.r.t. these variables, we have

$$\begin{aligned} \ln \pi_d^{(n)} = & \sum_{k=1}^{d-1} [\psi(\beta_k) - \psi(\alpha_k + \beta_k)] \\ & + \psi(\alpha_d) - \psi(\alpha_d + \beta_d) - \frac{a}{2b} \Delta_d(x^{(n)}), \end{aligned} \tag{33}$$

$$\pi_d^{(n)} \leftarrow \frac{\pi_d^{(n)}}{\sum_{l=0}^{\infty} \pi_l^{(n)}}, \tag{34}$$

where $\psi(\cdot)$ is the di-gamma function. This follows from a well-known formula [20]

$$\int \mathrm{d}v_d \, \mathrm{Beta}(v_d | \alpha_d, \beta_d) \ln v_d = \psi(\alpha_d) - \psi(\alpha_d + \beta_d). \tag{35}$$

Equations (26)-(32) and (33)-(34) are iteratively computed until convergence.

E. Deriving predictive distribution

Now we are ready to derive the predictive distribution (8). By definition, it is formally written as

$$p(x \mid \mathcal{D}') = \sum_{\boldsymbol{h}} \int \mathrm{d}\boldsymbol{\theta} \int \mathrm{d}\lambda \; p(x \mid \boldsymbol{h}, \boldsymbol{\theta}, \lambda) q(\boldsymbol{h}) q(\boldsymbol{\theta}) q(\lambda).$$

One problem here is that there is no explicit expression for the posterior $q(\boldsymbol{h})$ for an arbitrary value of x. For this, we use the following approximation. Imagine we had an augmented data set $\mathcal{D}' \cup x$, and we got a posterior on this $N+1$ data as

$$p_{\text{post.}}(\boldsymbol{h}, \mathsf{H}, \Psi \mid \mathcal{D}', x) = p(\boldsymbol{h}, \mathsf{H} \mid \Psi, \mathcal{D}', x) \; p(\Psi \mid \mathcal{D}', x), \tag{36}$$

where Ψ collectively represents $\boldsymbol{\theta}, \lambda, \boldsymbol{v}$. Equation (22) suggests that $p(\boldsymbol{h}, \mathsf{H} | \Psi, \mathcal{D}', x)$ should be factorized as

$$\begin{aligned} p(\boldsymbol{h}, \mathsf{H} \mid \Psi, \mathcal{D}', x) &= p(\boldsymbol{h} \mid \Psi, \mathcal{D}', x) \; p(\mathsf{H} \mid \Psi, \mathcal{D}', x) \\ &= p(\boldsymbol{h} \mid \Psi, x) \; p(\mathsf{H} \mid \Psi, \mathcal{D}'). \end{aligned} \tag{37}$$

The second line follows from the fact that the dependency of $\pi_d^{(n)}$ on $\mathcal{D}'$ is only through Ψ except for $x^{(n)}$. In Eq. (36), we can approximate as $p(\Psi \mid \mathcal{D}', x) \approx p(\Psi \mid \mathcal{D}')$ as long as $N \gg 1$, so that Ψ in $p(\boldsymbol{h} \mid \Psi, x)$ can be thought of the one learned from the original N sample data $\mathcal{D}'$. Therefore, we conclude that the posterior distribution of $\boldsymbol{h}$ is the categorical distribution whose d-th probability mass is given by

$$\begin{aligned} \ln \pi_d(x) = & \sum_{k=1}^{d-1} [\psi(\beta_k) - \psi(\alpha_k + \beta_k)] \\ & + \psi(\alpha_d) - \psi(\alpha_d + \beta_d) - \frac{a}{2b} \Delta_d(x), \end{aligned} \tag{38}$$

$$\pi_d(x) \leftarrow \frac{\pi_d(x)}{\sum_{l=0}^{\infty} \pi_l(x)}. \tag{39}$$

Using this approximation, we get

$$\begin{aligned} p(x \mid \mathcal{D}') \approx & \sum_{d=0}^{\infty} \pi_d(x) \int \mathrm{d}\boldsymbol{\theta} \; \mathcal{N}(\boldsymbol{\theta} | \boldsymbol{m}, \mathsf{\Sigma}) \\ & \times \sqrt{\frac{a}{b}} \mathcal{S} \left(\sqrt{\frac{a}{b}} (x - \boldsymbol{\theta}^\top \boldsymbol{\phi}_d) \middle| 2a \right) \\ \approx & \sum_{d=0}^{\infty} \pi_d(x) \; \mathcal{N} \left(x \middle| \boldsymbol{m}^\top \boldsymbol{\phi}_d, \frac{b}{a-1} + \boldsymbol{\phi}_d^\top \mathsf{\Sigma} \boldsymbol{\phi}_d \right), \end{aligned} \tag{40}$$

where $\mathcal{S}$ is Student's t-distribution (see Appendix). The last expression of Eq. (40) follows from the fact that the t distribution is approximated by Gaussian when the degrees of freedom is large. In this case, a is a large number on the order of N (see Eq. (28)), and this approximation is almost always justified.

Equation (40) also shows that the variance σ_d^2 in Eq. (8) is given by

$$\sigma_d^2 = \frac{b}{a-1} + \boldsymbol{\phi}_d^\top \mathsf{\Sigma} \boldsymbol{\phi}_d. \tag{41}$$

Since $\mathsf{\Sigma}$ is positive semidefinite, we see that σ_d^2 monotonically increases as $d = 0, 1, 2, \ldots$.

To implement the algorithm, we specify the maximum number of vehicles, D, which should be large enough depending the range of the count expected. Algorithm 1 summarizes the probabilistic vehicle counting algorithm.

III. Network flow estimation

Using Algorithm 1, we can estimate the vehicle count at any instant. However, this is only for the locations where webcams exist. Since the number of cameras is always much smaller than the number of links in the road network, we need a technology for extrapolation to monitor and manage the traffic over an entire city. Our goal is to estimate the traffic volume at arbitrary links of the network, given the observed traffic volume at a limited number of the links, as illustrated in Fig. 2.

A. Inverse Markov chain problem

We formalize this problem as an inverse Markov chain problem: Given the traffic volume at a limited number of the links, find the Markov transition probability $p(i|j)$, which is

Algorithm 1 Unsupervised counting.

```
I. Predictive distribution.
Input: Hyper-parameters $m_0, \Sigma_0, a_0, b_0, \beta$. Maximum count $D$.
Algorithm:
Initialize as $\pi_d^{(n)} = \frac{1}{D}, m = m_0, \Sigma = \Sigma_0$.
repeat
    Compute Eqs. (26)-(32) for $\{\Delta_d, a, b, \Sigma, m, \alpha_d, \beta_d\}$.
    Compute Eqs. (33)-(34) for $\{\pi_d^{(n)}\}$.
until Convergence.
Insert converged parameters into Eq. (38)-(40).
Return: Predictive distribution $p(x \mid \mathcal{D}')$.

II. Counting.
Input: Predictive distribution. New observation $x'$.
Algorithm:
Solve Eq. (10).
Return: Count $d'$.
```

defined as the transition probability from an arbitrary link j to another arbitrary link i.

Assume the Markov chain is irreducible meaning that completely isolated areas are not included in the map and any link is reachable from another link. Any irreducible Markov chain has a stationary distribution. Let the stationary distribution of this Markov chain be $s(i), i = 1, \dots, L$, where L denotes the total number of links in the network. Our fundamental assumption is that the observed traffic volume is proportional to $s(i)$ up to a measurement error:

$$y(i) = cs(i), \quad \forall i \in \mathcal{C}, \tag{42}$$

where $\mathcal{C}$ is the set of links directly monitored by webcams, $y(i)$ denotes the observed traffic volume for the i-th link (typically estimated from the approach in the previous section), and c is an unknown constant to be determined. Obviously, p and s satisfy

$$s(i) = \sum_{j=1}^{L} p(i \mid j)s(j) \tag{43}$$

which is also the definition of the stationary state probability. In the matrix form, this equation is written as $\mathsf{P}\boldsymbol{s} = \boldsymbol{s}$ in the obvious notation. This means that the stationary state is computed as the eigenvector of P having the eigenvalue of 1.

Here is the high-level procedure of the traffic flow estimation problem. Starting from Eq. (42), which holds only at the links having observed data, we solve the inverse Markov chain problem to get $p(i|j)$ for arbitrary pairs of links. Then we re-compute s using Eq. (43), which is done though eigen-decomposition of the probability matrix P, to recover the traffic volume at arbitrary links with and without observed data.

B. Parameterizing the transition model

We parameterize the probability distribution $p(i|j)$ as

$$p(i|j) = (1-\gamma)q(i \mid j\,;\, \boldsymbol{u}) + \gamma r(i;\, \boldsymbol{w}) \tag{44}$$

where γ is called the restart probability (assumed to be a fixed parameter), and $\boldsymbol{u}$ and $\boldsymbol{w}$ are the model parameters to be learned. In this decomposition, r is interpreted as the initial probability distribution over the links, while q is interpreted as the "partial" transition probability distribution. This type of decomposition is natural for traffic analysis on road networks since it is consistent with a typical data generation process in traffic simulation. Specifically, when we generate traffic data using a multi-agent simulator [21], we first generate the starting locations and then generates paths according to a given transition rule.

For r and q, we use the following particular forms:

$$r(i;\, \boldsymbol{w}) \propto \exp(w_i) \tag{45}$$

$$q(i \mid j\,;\, \boldsymbol{u}) \propto I(i \sim j)\exp[g(\boldsymbol{u})] \tag{46}$$

$$g(\boldsymbol{u}) \equiv u_{i,j} + u_0 \cos(i|j) + u_1 h_{\mathrm{type}(i)} \tag{47}$$

where $i \sim j$ represents that the i-th link is directly connected to the j-th link, and $I(\cdot)$ is the indicator function. Notice that the transition probability between unconnected links is zero by definition. Unlike the conventional approach, which imposes the traffic conservation constraint directly on the flow itself, we take account of the conservation law *probabilistically*. Once the transition probability is found, the normalization condition of the transition probability automatically guarantees the conservation law in the expectation.

In Eq. (47), $\cos(i|j)$ is the cosine of the geometric angle between the i-th and j-th links. For example, if the j-th link points in the opposite direction to the i-th link, $\cos(i|j) = -1$ and the transition probability between them is down-weighted. If they point in the same direction, then that transition should occur more often. The term $u_{i,j}$ is a correction term to the cosine similarity, which is expected to play a minor role (initialized to zero in optimization). We believe that this is a generally acceptable model, but the term $u_{i,j} + u_0 \cos(i|j)$ can be replaced with another link-link similarity if more detailed knowledge is available.

The last term of Eq. (47) captures the variability in the road type. Specifically, the function $\mathrm{type}(i)$ returns the road type index for the i-th link. Based on the link attributes available in a digital road map, we define 14 road types including motorway, trunk, primary, secondary, etc., as listed in Table I (see Section V-B), and each of them are weighted differently with h_t $(t = 1, \dots, 14)$.

Combining Eqs. (44)-(46), we obtain the stationary distribution $s(i)$ as a function of the model parameters $\boldsymbol{w} = [w_0, \dots, w_L]^\top$ and $\boldsymbol{u} = [u_0, u_1, u_{1,1}, \dots, u_{L,L}]^\top$, where $u_{i,j}$'s for unconnected pairs are omitted. Optimal model parameters are those that minimize the discrepancies between the left and right hand sides of Eq. (42). The key question is how to measure the discrepancy, which will be discussed in the next subsection.

C. Designing the objective function

Now that we have introduced $s(i)$ as a probability distribution, Eq. (42) can be viewed as a relationship between two distributions. The most natural discrepancy measure for distributions is the Kullback-Leibler (KL) divergence [22], which has been used in a number of traffic estimation problems [23], [24], [25]. The KL divergence can be interpreted as the expectation of information loss. Let us define $\rho(i)$ by

$$\rho(i) \equiv \ln \frac{cs(i)}{y(i)}. \tag{48}$$

Then $\rho(i)$ represents the local information loss at link i. Since we are interested in the stationary state, the information loss on the network should be as uniform as possible. If there is a large loss at a particular link, it will be dissipated through the transition process. With this intuition in mind, we define the error function to be minimized as the variance of the information loss:

$$L(\boldsymbol{u}, \boldsymbol{w}) \equiv \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \left[\rho(i) - \bar{\rho}\right]^2, \tag{49}$$

where $|\mathcal{C}|$ is the number of links directly monitored by the webcams, and $\bar{\rho}$ is the mean of the information loss defined by

$$\bar{\rho} \equiv \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \rho(i). \tag{50}$$

Using the definition of ρ, we obtain the final expression of the error function as

$$L(\boldsymbol{u}, \boldsymbol{w}) = \frac{1}{2|\mathcal{C}|^2} \sum_{i,j \in \mathcal{C}} \left[\ln \frac{cs(i \mid \boldsymbol{u}, \boldsymbol{w})}{y(i)} - \ln \frac{cs(j \mid \boldsymbol{u}, \boldsymbol{w})}{y(j)}\right]^2 \tag{51}$$

after some algebra, where we explicitly represented the dependency on the model parameters $\boldsymbol{u}$ and $\boldsymbol{w}$ in $s(i)$ and $s(j)$. We see that the unknown c is canceled in this objective.

In addition, we impose an elastic net type regularization [26] on the model parameters:

$$R_1(\boldsymbol{u}, \boldsymbol{w}) \equiv |u_0| + |u_1| + \sum_{i \sim j} |u_{i,j}| + \sum_{i=1}^{L} |w_i| \tag{52}$$

$$R_2(\boldsymbol{u}, \boldsymbol{w}) \equiv {u_0}^2 + {u_1}^2 + \sum_{i \sim j} {u_{i,j}}^2 + \sum_{i=1}^{L} {w_i}^2. \tag{53}$$

By putting all together, the final objective function to be minimized is given by

$$Q(\boldsymbol{u}, \boldsymbol{w}) \equiv L(\boldsymbol{u}, \boldsymbol{w}) + \lambda_1 R_1(\boldsymbol{u}, \boldsymbol{w}) + \lambda_2 R_2(\boldsymbol{u}, \boldsymbol{w}), \tag{54}$$

where the new parameters λ_1 and λ_2 control the tradeoff between the error function and the regularization terms, and are determined through cross validation. As previously mentioned, the flow conservation constraint is probabilistically considered in our model.

D. Solving the optimization problem

The objective function $Q(\boldsymbol{u}, \boldsymbol{w})$ can be minimized with the gradient method. We developed an efficient algorithm based on the notion of natural gradient [27], but details are omitted here due to space limitations. For the details of the gradient method, see a companion paper [17].

Once the minimizer $\boldsymbol{u}^*$,$\boldsymbol{w}^*$ of $Q(\boldsymbol{u}, \boldsymbol{w})$ is found, one can determine an optimal c (denoted by c^*) by solving the least squares problem:

$$c^* = \arg\min_c \sum_{j \in \mathcal{C}} \left[y(j) - cs(j \mid \boldsymbol{u}^*, \boldsymbol{w}^*)\right]^2 \tag{55}$$

to get the solution

$$c^* = \left[\sum_{i \in \mathcal{C}} s(i \mid \boldsymbol{u}^*, \boldsymbol{w}^*)^2\right]^{-1} \sum_{j \in \mathcal{C}} y(j) s(j \mid \boldsymbol{u}^*, \boldsymbol{w}^*). \tag{56}$$

Finally, the network inference algorithm is summarized as follows:

Algorithm 2 Inverse Markov network traffic inference.

```
Input: Observed traffic flows ${y(i) | i ∈ \mathcal{C}}$. Restart probability $\gamma$. Regularization parameters $\lambda_1, \lambda_2$.
Algorithm:
  • $(\boldsymbol{u}^*, \boldsymbol{w}^*) = \arg\min_{\boldsymbol{u},\boldsymbol{w}} Q(\boldsymbol{u}, \boldsymbol{w})$.
  • Find the transition probability matrix $\mathsf{P}$ using Eq. (44).
  • Find the stationary distribution $\boldsymbol{s}$ by solving $\mathsf{P}\boldsymbol{s} = \boldsymbol{s}$.
  • Compute $c^*$ using Eq. (56).
  • Compute $\hat{\boldsymbol{y}} = c^* \boldsymbol{s}$.
Return: Estimated traffic flow $\{\hat{y}(j) \mid \forall j\}$.
```

IV. Related work

This section reviews related work with an emphasis on the task of image-based traffic estimation and network-wide traffic estimation.

A. Image-based traffic estimation

In transportation research, the cost issue of ITS deployment is one of the major recent concerns. Replacing the traditional data acquisition infrastructures with alternative methods is a common approach. Mainly two alternative methods have been studied to date: GPS (global positioning system) and surveillance cameras.

The use of surveillance cameras can viewed as the mainstream of lightweight ITS development. One of the pioneering work is a webcam-based ITS proposed by Santini [13], which attempts to solve the OD matrix estimation problem based on partial observation given by image analysis. Although it shares the main motivation with ours, it is unable to estimate the traffic volume at arbitrary links. Yu et al. [28] presents another early study on a lightweight ITS using network-connected cameras. The advantage of webcams in terms of deployment costs is also described by Huck et al. [29] based on their own implementation.

Although camera images provide rich enough information for traffic monitoring, one of the major problems with the use of general-purpose cameras is the lack of capability of calibration. Cathey and Dailey [3] proposed a sophisticated algorithm to estimate traffic speed based on cross-correlation without calibrations on the camera side. Tian et al. [5] and Buch et al. [6] describe how to enable vehicle recognition

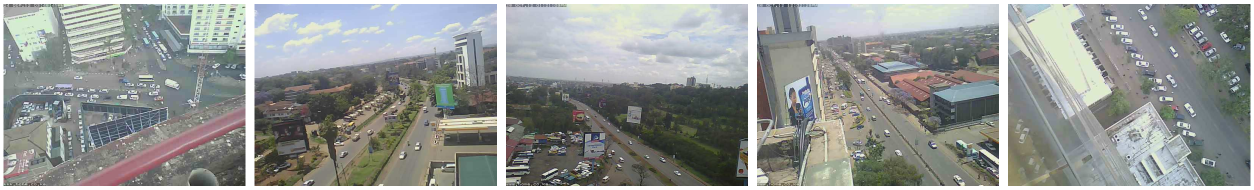

Fig. 5. Examples of original webcam images. From left to right, Nationkimathi, Westistg, Ukulima, Haileselasie, and Harambeetaifa, in Nairobi City [2].

from low-resolution images. Robert [4] attempts to enhance the accuracy of vehicle recognition by incorporating machine learning algorithms such as support vector machines. Once vehicle recognition is done, the task of vehicle counting is trivial.

As discussed in Introduction, object recognition from low-quality images is a challenging task in general. One serious bottleneck in practice is the cost of preparing properly labeled images. Although a recently proposed regression-based approach [7] removed the image recognition step and reduced the task to supervised learning, a fair amount of manually labeled training data is still required. From this perspective, a congestion prediction approach proposed by Porikli and Xiaokun [30] is quite interesting. They trained a hidden Markov model with *unlabeled* data, and associated the learned state sequence with a decision rule for congestion. The input feature is automatically generated via the discrete cosine transform. Although their problem clearly differs from ours, they share the same spirit with us.

B. Network-wide traffic estimation

Once link-wise traffic estimation is done at the links being monitored, the next task is to extrapolate the observation to the entire network. Since the advent of a seminal work by Zuylen and Willumsen [9], the task of network-level traffic estimation from limited observations has attracted a lot of attention. Most of the previous efforts have focused on the task of origin-destination (OD) matrix estimation [10], [11], where the following two approaches have been mainly studied.

The first approach is to minimize an error function between observed and estimated traffic volumes while satisfying flow conservation conditions. Recent work includes Shao et al. [12], which estimates the OD matrix from partial observations under dynamic traffic variations. Hu et al. [31] addressed essentially the same task but in a different context of sensor fusion. Although our approach shares some of the ideas with these, it can be clearly distinguished from them in that our goal is not to estimate the OD matrix but the traffic volume at arbitrary links. Also, from a mathematical perspective, we leverage the regularization theory developed in machine learning. In particular, we leverage the elastic net regularization [26] to achieve both the sparsity and numerical stability at the same time. We also use a novel cost function inspired by an information-theoretic interpretation. Menon et al. [25] recently introduced an interesting algorithm for sparse OD matrix estimation using an L_1 regularization technique. Their problem as well as the optimization strategy differs from ours as clearly represented by the fact that the "seed" OD matrix is not available in our problem.

The other major approach is to use Bayesian network, where graphical Gaussian models (GGMs) [32] are commonly used. For instance, Zhang et al. [33] used GGMs for short-term traffic forecasting. Sun et al. [34] extended the model to include multiple Gaussian components although the resulting probabilistic model is no longer a Bayesian network. Chen et al. [35] introduced an approach to estimate the OD matrix or the link traffic matrix based on the explicit expression of the conditional distribution of GGMs. Zhu et al. [36] also used a GGM for the network sensor location problem, where a loss function based on the trace of the covariance matrix is proposed. Although these approaches are built upon the well-grounded theory of GGMs, and thus easier to analyze, one of the practical disadvantages is the scalability. For large networks, the global Gaussian assumption is hard to apply. Also, it is well known that the naive use of GGMs faces a serious computational issue in high-dimensional systems [37]. Thanks to a carefully designed regularization term and the natural gradient algorithm [27], [17], our method easily scales to networks of thousands of links. In the next section, we will show experimental results on a real road network in Nairobi, Kenya.

V. Experiments

We prototyped a traffic monitoring system using existing webcams in Nairobi, provided by AccessKenya.com [2]. In the downtown area of Nairobi, there are 1 497 links, while only 52 links are monitored by the webcams (about 3.5%).

A. Vehicle counting

Figure 5 shows images from the webcams at five major locations being monitored. As seen, the webcams are typically mounted on buildings. We see that they are quite far from the roads. Although the original size of the images are 640×480, the number of pixels in the region of interest is just several hundred as suggested by Fig. 3.

For those locations, we generated $N = 100$ images by randomly picking still images at different times over several days. We use the relative mean absolute error (RMAE) for performance comparison:

$$\mathrm{RMAE} = \frac{1}{100} \sum_{n=1}^{100} \frac{\left| d_{\mathrm{true}}^{(n)} - d_{\mathrm{estimate}}^{(n)} \right|}{d_{\mathrm{true}}^{(n)} + 1}. \qquad (57)$$

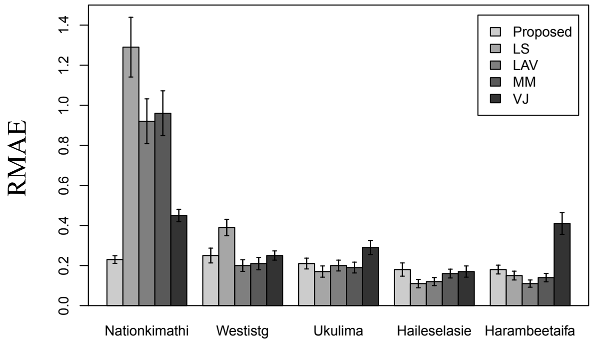

Fig. 6. Comparison of the relative mean absolute error (RMAE).

Here $d_{\rm true}$ is the ground truth count, which was manually prepared spending several person-day workloads. We use the leave-one-out cross validation scheme to compute RMAE.

We compared our unsupervised approach with *supervised* alternatives. We used linear regression methods of least squares linear regression (LS), least absolute values (LAV), and MM estimator (MM). See [38] for details of these algorithms. To train those, we used the true count labels in addition to the vehicle-pixel-area feature, and hence the comparison is extremely preferable to the alternatives. We did not use nonlinear regression methods such as Gaussian process regression [7], because our preliminary experiments showed that the vehicle pixel area feature is mostly linearly correlated with the count.

We also compared with a widely used object recognition approach by Viola and Jones (VJ) [39]. To make it work, we gave several hundred manually labeled images from the webcams in our setting in addition to 2 000 labeled images with positive (vehicle) and negative (non-vehicle) labels from general image databases containing vehicles [40], [41], [42], [43]. Thus in terms of the cost to prepare the training data, the following inequality holds:

$$(\text{proposed method}) \lll (\text{LS}, \text{LAV}, \text{MM}) \ll \text{VJ}. \tag{58}$$

We remind the reader again that the proposed method does not need any labeled data for training.

Figure 6 shows RMAE values, where the error bars represent the standard deviation. The initial parameters are $\boldsymbol{m}_0 = (-1, 0.3)^\top, \Sigma_0 = 10^{10}\mathsf{I}_2, a_0 = 1, b_0 = 10^{10}$, and $D = 100$, where I_2 is the two-dimensional identity matrix. Regarding β, we put a non-informative hyper-prior on it for numerical stability. This leads to a slight modification of the VB updating equations. For the details, see our companion paper [16]. The proposed VB algorithm took only a few seconds on a moderate laptop computer. The time complexity is $O(N)$. The figure shows that proposed method is comparable to or even better than the supervised alternatives in terms of the error as well as robustness. In particular, when the resolution of images very low as is the case in Nationkimathi, our method clearly outperforms the supervised alternatives.

Figure 7 shows the predictive distribution for the images of Nationkimathi. To plot this, we used the sample average for the mixture weight as $\bar{\pi}_d \equiv \frac{1}{N}\sum_{n=1}^{N} \pi_d(x^{(n)})$ to see the general traffic status of this location. We see that in most of time the count is zero there, while $d = 5$ seems to be another commonly observed situation.

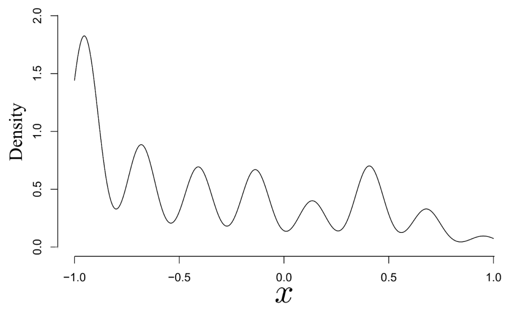

Fig. 7. Averaged predictive distribution at Nationkimathi.

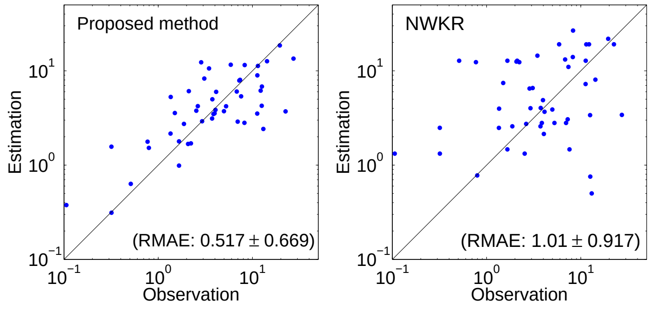

Fig. 8. Comparison of RMAE in the network flow.

B. Network inference

Using the estimated traffic volume using the vehicle counting algorithm, we estimated city-wide traffic volume at arbitrary links by solving the inverse Markov chain problem. In the Nairobi data, $|\mathcal{C}| = 52$ links are monitored by the webcams, while the total number of links is $L =$1 497.

For our approach, we fixed $\lambda_1 = 1$, and initialized as $\boldsymbol{w} = \boldsymbol{0}, u_0 = u_1 = 1, u_{i,j} = 0$. For the road-type weight h, we used the values listed in Table I multiplied by $\ln[1 + N_{\rm L}]$, $N_{\rm L}$ being the number of lanes of the road. For λ_2 and γ, we used cross-validation to choose the values. Thanks to the $\rm L_1$ regularizer, more than 70% of the entries of $\boldsymbol{w}, \boldsymbol{u}$ became zero after optimization.

We compared our method with Nadaraya-Watson kernel regression (NWKR), where the flow of an arbitrary link is estimated simply as a linear combination of the observed values, and the coefficients (kernel functions) are computed

TABLE I
ROAD TYPE WEIGHT (h) FOR EQ. (47).

motorway	1.5	secondary	0.3
motorway_link	1.3	secondary_link	0.1
trunk	1.1	tertiary	−0.1
trunk_link	0.9	tertiary_link	−0.3
primary	0.7	unclassified	−0.5
primary_link	0.5	other	−0.7

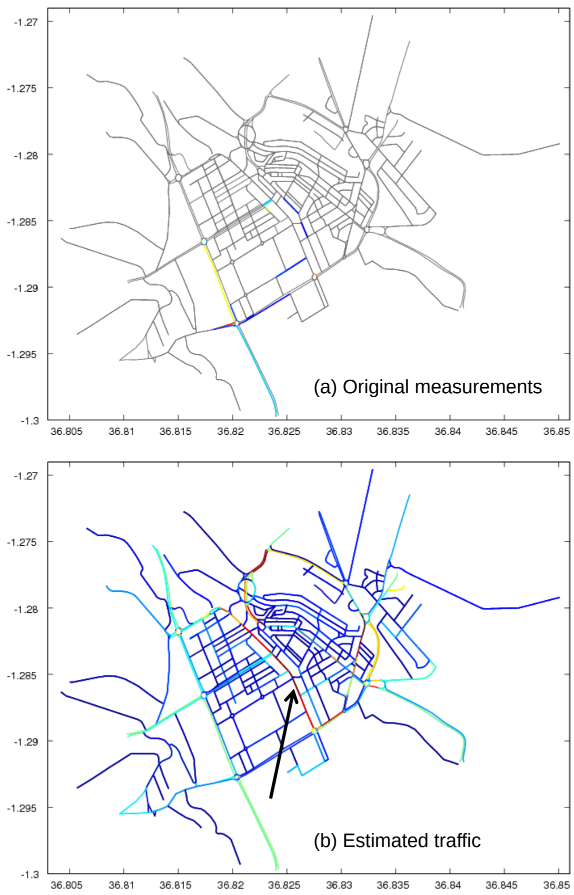

Fig. 9. Network flow estimation results in downtown Nairobi.

based on the number of hops from the i-th to the j-th links, $N(j|i)$, in the road network [44]:

$$s_{\mathrm{NWKR}}(j) = \frac{\sum_{i=1}^{|\mathcal{C}|} \mathrm{e}^{-\alpha N(j|i)} y(i)}{\sum_{i=1}^{|\mathcal{C}|} \mathrm{e}^{-\alpha N(j|i)}},$$

where α is the parameter to be determined using cross validation.

Figure 8 shows the results. To compare between estimation and observation, we used leave-one-out cross validation over the 52 observed links. In these figure, the 45° line represents perfect agreement. As seen, our method gives much better agreement than NWKR. The figures also show values of RMAE. In terms of RMAE, our method is about twice better than the alternative.

Figure 9 compares the original and estimated traffic in color in downtown Nairobi. The red and yellow roads are most congested, while the traffic on the blue roads is flowing smoothly. In Nairobi, traffic congestion in the downtown is a serious social problem, as pointed out by a local traffic survey report [45]. The most congested road highlighted with the arrow was in fact consistent to the survey.

VI. Conclusion

We have proposed a new approach to ITS. Our system consists of two major functionalities: (1) webcam-based traffic monitoring and (2) city-wide network flow estimation. The traffic monitoring module features a new algorithm for computing the vehicle counts from very low-resolution webcam images. The major feature is that it does not require any labeled (i.e. manually counted or recognized) images. For the network flow estimation module, we formalize the problem as an inverse Markov chain problem, and reduce it to a regularized optimization problem. Using real webcams deployed in Nairobi, Kenya, we demonstrated the practical utility of our approach.

Appendix A
Probability distributions

The definition of the gamma, beta, Gaussian, and Student's t distributions is given as follows:

$$\mathrm{Gam}(\lambda \mid a, b) \equiv \frac{b^a}{\Gamma(a)} \lambda^{a-1} \mathrm{e}^{-b\lambda},$$

$$\mathrm{Beta}(v \mid \alpha, \beta) \equiv \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} v^{\alpha-1}(1-v)^{\beta-1},$$

$$\mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{m}, \Sigma) \equiv \frac{|\Sigma|^{-\frac{1}{2}}}{(2\pi)^{\frac{W}{2}}} \exp\left\{ -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^\top \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{m}) \right\},$$

$$\mathcal{S}(z \mid \nu) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi\nu}\,\Gamma(\frac{\nu}{2})} \left(1 + \frac{z^2}{\nu}\right)^{-\frac{\nu+1}{2}},$$

where $\Gamma(\cdot)$ is the gamma function and W is the dimensionality of $\boldsymbol{\theta}$.

References

[1] N. Radjou, J. Prabhu, and S. Ahuja, *Jugaad Innovation*. Jossey-Bass, 2012.

[2] AccessKenya.com, "http://traffic.accesskenya.com/."

[3] F. Cathey and D. Dailey, "A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras," in *Proceedings of the IEEE Intelligent Vehicles Symposium, 2005*, 2005, pp. 777–782.

[4] K. Robert, "Video-based traffic monitoring at day and night vehicle features detection tracking," in *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems*, ser. ITSC 09, 2009, pp. 1–6.

[5] B. Tian, Q. Yao, Y. Gu, K. Wang, and Y. Li, "Video processing techniques for traffic flow monitoring: A survey," in *Proceedings of the IEEE 14th International Conference on Intelligent Transportation Systems, 2011*, 2011, pp. 1103–1108.

[6] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Transactions on Intelligent Transportation System*, vol. 12, no. 3, pp. 920–939, 2011.

[7] M. Liang, X. Huang, C. H. Chen, X. Chen, and A. Tokuta, "Counting and classification of highway vehicles by regression analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2878–2888, 2015.

[8] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascón, and D. Oñoro-Rubio, "Extremely overlapping vehicle counting," in *Proceedings of the 7th Iberian Conference on Pattern Recognition and Image Analysis*, ser. Lecture Notes in Computer Science, vol. 9117, 2015, pp. 423–431.

[9] H. J. V. Zuylen and L. G. Willumsen, "The most likely trip matrix estimated from traffic counts," *Transportation Research Part B: Methodological*, vol. 14, no. 3, pp. 281–293, 1980.

[10] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: Recent developments," *Statistical Science*, vol. 19, no. 3, pp. 499–517, 2004.

[11] E. Lawrence, G. Michailidis, V. N. Nair, and B. Xi, "Network tomography: A review and recent developments," in *In Fan and Koul, editors, Frontiers in Statistics*, 2006, pp. 345–364.

[12] H. Shao, W. H. Lama, A. Sumalee, A. Chen, and M. L. Hazelton, "Estimation of mean and covariance of peak hour origin-destination demands from day-to-day traffic counts," *Transportation Research Part B: Methodological*, vol. 68, pp. 52–75, 2014.

[13] S. Santini, "Analysis of traffic flow in urban areas using Web cameras," in *Proceedings of IEEE Workshop on Applications of Computer Vision*, 2000, pp. 140–145.

[14] C.-N. Anagnostopoulos, I. Anagnostopoulos, I. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 377–391, 2008.

[15] T. Idé, T. Katsuki, T. Morimura, and R. Morris, "Monitoring entire-city traffic using low-resolution Web cameras," in *Proceedings of the 20th ITS World Congress, Tokyo*, 2013.

[16] T. Katsuki, T. Morimura, and T. Idé, "Bayesian unsupervised vehicle counting," in *IBM Research Report, RT0951*, 2013.

[17] T. Morimura, T. Osogami, and T. Idé, "Solving inverse problem of Markov chain with partial observations," in *Advances in Neural Information Processing Systems*, 2013, pp. 1655–1663.

[18] N. Otsu, "A threshold selection method from gray-level histogram," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, pp. 62–66, 1979.

[19] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[20] Wikipedia, "https://en.wikipedia.org/wiki/Beta_distribution."

[21] T. Osogami, T. Imamichi, H. Mizuta, T. Suzumura, and T. Idé, "Toward simulating entire cities with behavioral models of traffic," *IBM Journal of Research and Development*, vol. 57, no. 5, pp. 6:1–6:10, 2013.

[22] S. Kullback and R. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 55, pp. 79–86, 1951.

[23] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM 03, 2003, pp. 301–312.

[24] N. Shlayan, P. Kachroo, and S. Wadoo, "Transportation reliability based on information theory," in *Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems*, ser. ITSC 11, 2011, pp. 1415–1420.

[25] A. K. Menon, C. Cai, W. Wang, T. Wen, and F. Chen, "Fine-grained od estimation with automated zoning and sparsity regularisation," *Transportation Research Part B: Methodological*, vol. 80, pp. 150–172, 2015.

[26] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society*, vol. 67, pp. 301–320, 2005.

[27] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[28] X. D. Yu, L. Y. Duan, and Q. Tian, "Highway traffic information extraction from skycam MPEG video," in *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems, 2002*, 2002, pp. 37 – 42.

[29] R. Huck, J. Havlicek, J. Sluss, and A. Stevenson, "A low-cost distributed control architecture for intelligent transportation systems deployment in the State of Oklahoma," in *Proceedings of the 2005 IEEE Intelligent Transportation Systems*, 2005, pp. 919–924.

[30] F. Porikli and L. Xiaokun, "Traffic congestion estimation using HMM models without vehicle tracking," in *2004 IEEE Intelligent Vehicles Symposium*, 2004, pp. 188–193.

[31] S.-R. Hu, S. Peeta, and H.-T. Liou, "Integrated determination of network origin-destination trip matrix and heterogeneous sensor selection and location strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 195–205, 2016.

[32] S. L. Lauritzen, *Graphical Models*. Oxford, 1996.

[33] C. Zhang, S. Sun, and G. Yu, "A Bayesian network approach to time series forecasting of short-term traffic flows," in *Proceedings of IEEE International Conference on Intelligent Transportation System*, 2004, pp. 216–221.

[34] S. Sun, C. Zhang, and G. Yu, "A Bbayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.

[35] L. Cheng, S. Zhu, Z. Chu, and J. Cheng, "A Bayesian network model for origin-destination matrices estimation using prior and some observed link flows," *Discrete Dynamics in Nature and Society*, vol. Article ID 192470, 2014.

[36] S. Zhu, L. Cheng, Z. Chu, A. Chen, and J. Chen, "Identification of network sensor locations for estimation of traffic flow," *Transportation Research Record*, no. 2443, pp. 32–39, 2014.

[37] M. Drton and M. D. Perlman, "A SINful approach to Gaussian graphical model selection," *Journal of Statistical Planning and Inference*, vol. 138, pp. 1179–1200, 2008.

[38] R. R. Wilcox, *Introduction to robust estimation and hypothesis testing*. Academic Press, 2012.

[39] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ser. CVPR 01, vol. 1. IEEE, 2001, pp. 511–518.

[40] P. Negri, X. Clady, S. M. Hanif, and L. Prevost, "A cascade of boosted generative and discriminative classifiers for vehicle detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, p. 136, 2008.

[41] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, jun 2010.

[42] ——, "The PASCAL VOC2012 Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[43] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.

[44] T. Idé and S. Kato, "Travel-time prediction using Gaussian process regression: A trajectory-based approach," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, ser. SDM 09, 2009, pp. 1185–1196.

[45] E. J. Gonzales, C. Chavis, Y. Li, and C. F. Daganzo, "Multimodal transport in Nairobi, Kenya: Insights and recommendations with a macroscopic evidence-based model," in *Transportation Research Board 90th Annual Meeting*, 2011, pp. 11–3045.

Tsuyoshi Idé Tsuyoshi Idé is a Senior Technical Staff Member of IBM Research, IBM T. J. Watson Research Center. He received his Ph.D. from University of Tokyo in 2000 in theoretical solid-state physics. He joined IBM Research – Tokyo in 2000, and moved to T. J. Watson Research Center in 2012. His major research area is data mining and machine learning. E-mail: tide@us.ibm.com.

Takayuki Katsuki Takayuki Katsuki is a researcher at IBM Research - Tokyo. He received a M.E. degree in electrical engineering and bioscience from Waseda University, Tokyo, Japan, in 2012. His research interests include machine learning, data mining, and their applications. He has been awarded the IEEE Computational Intelligence Society, Japan Chapter, Young Researcher Award in 2011. E-mail: kats@jp.ibm.com.

Tetsuro Morimura Tetsuro Morimurais a researcher at IBM Research Tokyo. He received M.E. and Ph.D. degrees in engineering from the Nara Institute of Science and Technology, Japan, in 2005 and 2008. His research interests include reinforcement learning. E-mail: tetsuro@jp.ibm.com.

Robert Morris Robert J. T. Morris is Vice President of Global Laboratories, IBM Research, including labs in China, India, Japan, Australia, Brazil and Africa. Before the current role, he was VP of Services Research, IBM, T. J. Watson Research Center, the Director of the IBM Almaden Research Center, and the VP responsible for research in IBMs ThinkPad. He is a Fellow of the IEEE. E-mail: rjtm@us.ibm.com.