Monitoring Entire-City Traffic using Low-Resolution Web Cameras

Tsuyoshi Idé, Takayuki Katsuki, Tetsuro Morimura
IBM Research – Tokyo
{goodidea, kats, tetsuro}@jp.ibm.com

Robert Morris
VP of Global Labs, IBM Research
rjtm2@us.ibm.com

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 and velocity from very low-resolution Web camera images, while the network flow estimation module features a traffic flow estimation algorithm at every single link, based on measurements at a limited number of links with the cameras. Using real Web cameras deployed in Nairobi, Kenya, we assessed the accuracy of our approach. To the best of the authors' knowledge, this is the first practical framework for monitoring an entire city's traffic without special expensive infrastructure and time-consuming data calibration.

Keywords: Traffic Monitoring, Image Recognition, Network Analysis, Frugal Innovation

1. 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.

This paper presents such a "frugal" approach [1] to ITS. With Nairobi City, Kenya, for our prototype, we developed a traffic monitoring system for the entire city. A major feature of our system is that it imposes only a minimum cost beyond the road infrastructure itself. Instead of using traditional road-side sensors, the system takes advantage of the existing Web cameras of AccessKenya.com [5], most of which are mounted on commercial buildings and individual residences, and the traffic information is monitored through Web browsers. This system needs no expensive infrastructure, but we have to overcome several technical hurdles.

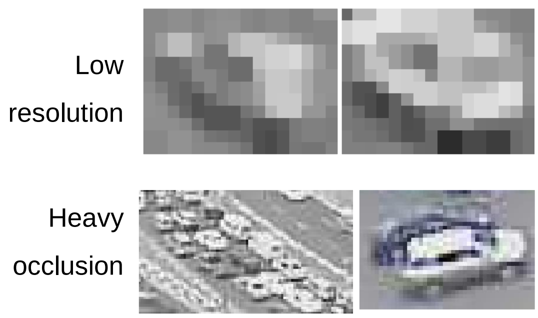

Fig 1. Examples of very low quality images

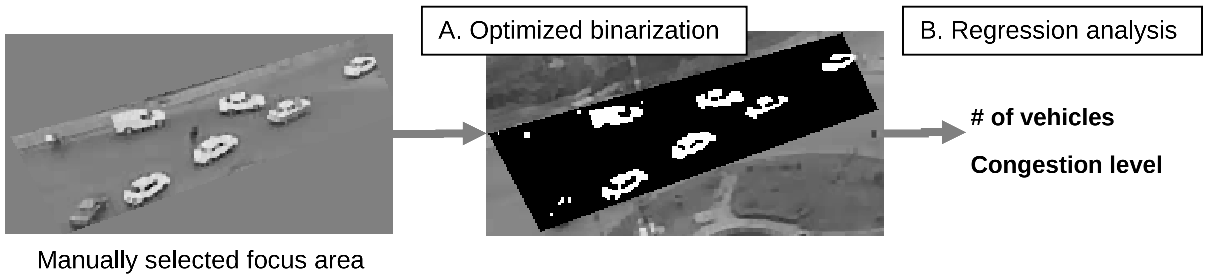

Fig.2. Outline of vehicle counting algorithm

The *first* challenge is how to handle Very Low-Quality Images (VLQI; see Fig.1 for examples). Due to cost and antitheft concerns, special-purpose close-view cameras are not suitable in most developing countries. In our system, the Web cameras are general-purpose cameras typically mounted on buildings, and their image quality is extremely low. Standard object recognition technologies such as those used in number plate recognition [3] are useless.

The *second* challenge is how to eliminate the time-consuming step of calibration in the image processing. Most of the existing video-based technologies for traffic monitoring require an elaborate design to select image features for object recognition. For example, Robert [2] presents a video-based vehicle tracking system, where image patches are compared with template images of windshields and headlights to identify vehicles. For accurate matching, each camera needs to undergo rather tedious calibration steps on camera positions and distances.

The *third* challenge is how to derive city-wide information from a limited number of Web cameras. In particular, what-if simulation for optimized city planning calls for estimating the traffic flow in every single link of the road network. This task is similar to network tomography, whose application to ITS was first proposed in [4], but differs in that we need to infer all of the link costs instead of just the origin-destination flows.

We tackle these challenges with sophisticated machine learning techniques. First, for the first and second challenges, instead of the standard template-matching approach, we propose to use a simple regression model combined with an optimal threshold for binarization. Our algorithm (in Section 2) allows accurately estimating the vehicle flow at the locations monitored by Web cameras even when the quality of the images is too low for template matching. Second, we develop a new inference method on road networks to estimate the traffic flow at an arbitrary link without direct observation by Web cameras. The idea is to take advantage of the correlations between links. For example, imagine there is only a single highway in a region, and the highway consists of a number of consecutive links. In this case, if the traffic flow for a single link is given, we readily see the flow of the neighboring links will be the same. For more complex network having many branching points, we can extend the idea by using a mathematical technique of *regularization* (described in Section 3). Finally, in Section 4, we show some experimental results for our prototype based on the actual traffic in Nairobi, Kenya.

2. CALIBRATION-FREE VERY LOW QUALITY IMAGE ANALYAIS

This section presents an approach to calibration-free VLQI analysis to estimate the traffic volume. Figure 2 shows an outline of our approach. We assume that a number of Web cameras are monitoring the traffic in a city. In our prototype in Nairobi, there are 27 Web cameras. For each Web camera, we manually define a region of interest (ROI) depending on the location and the distance to the link of interest, as highlighted by the black region in the figure. The ROI selection is performed only once, depending on user's preference, and requires no fine tuning.

Since the Web cameras are analyzed independently, we focus on a single Web camera in the rest of this section. We assume that we are given a training data set for the camera, which contains N images. Each of the images is assumed to have M pixels, and each of the pixels takes a value out of 256 levels representing luminance. As a preprocess, we subtract from the pixel values of each image the median of its M pixels to handle the variation in luminance. For example, there is considerable luminance difference between rainy and clear days. We denote the normalized pixel value by $z_i^{(n)}$ for the i-th pixel of the n-th image (i = 1, 2, …, M ; n = 1, 2, .., N). We also assume that each of the images is associated with the vehicle count, $y^{(n)}$ for the n-th image, which is assumed to be obtained by manual counting or other means. To summarize, our training data set can be represented as $\{ (y^{(n)}, z_1^{(n)}, \ldots, z_M^{(n)}) \mid n = 1, 2, .., N \}$.

2-1. Optimized binarization

As indicated in Fig. 2, the first step is to transform the normalized images into black-white images to highlight the individual vehicles. The rule for transformation (binarization) is simple: if the normalized pixel value is greater than a threshold, k, the pixel value is replaced with 1. Otherwise, it is set to 0. Now our task is to find the optimal binarization threshold k.

Figure 3 shows an example of the distribution of pixel values of the training set, where the red bar separates the black (called class 1) from the white (class 2) pixels. There are MN pixel values in the data. Let n_l be the frequency of the l-th level. By definition,

$$n_l = \sum_{n=1}^{N} \sum_{i=1}^{M} I(z_i^{(n)} = l),$$

and $n_0 + n_1 + \ldots + n_{255} = MN$. In this definition, $I(\cdot)$ is the indicator function that gives 1 when

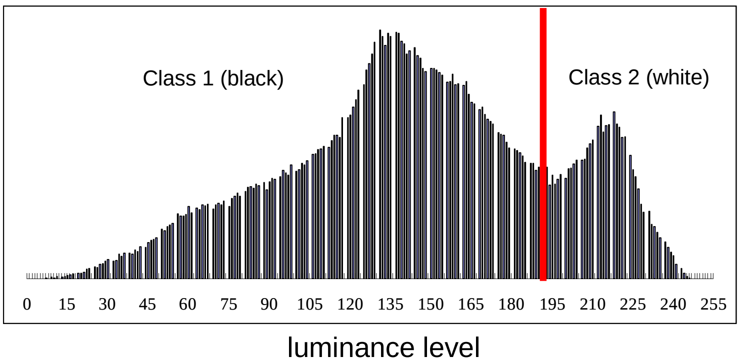

Fig. 3. Luminance distribution of a data set.

the argument is true, 0 otherwise. To determine the optimal threshold, we follow the method in [6]. The idea is to maximize the statistical variance between the white and the black pixels. Let $m_1(k)$ and $m_2(k)$ be the mean luminance values of the classes 1 and 2, respectively. Obviously, these are functions of the threshold k. Also, let m_T be the total mean. Then, the between-class variance s_B^2 is defined as the variance of a population with the mean m_T, which includes only two samples at $m_1(k)$ and $m_2(k)$. The value of s_B^2 will be large when the distribution has two distinct peaks and the threshold separates them very well, as shown in the figure. The optimal threshold k^* is found as

$$k^* = \arg\max_k s_B^2(k). \qquad (1)$$

A straightforward approach to this optimization problem is to evaluate $s_B^2(k)$ at 256 different values of k, and pick one giving the maximum.

2-2. Regression model

Given the optimized threshold k^*, our next step is to find the relationship between the vehicle count and the number of white pixels. Let $x^{(n)}$ be the number of white pixels in the n-th image, calculated based on the optimal threshold. Now our task is to find a functional relationship between y (vehicle count) and x (number of white pixels), based on the training set $\{ (x^{(n)}, y^{(n)}) \mid n = 1, \ldots, N\}$.

For this task, we propose a simple regression model. We assume a linear model between x and y as $y = ax + b$, and determine the parameters a and b, based on the training data. Formally, our problem is stated as

$$(a^*, b^*) = \arg\min_{a,b} \sum_{n=1}^{N} \left(y^{(n)} - ax^{(n)} - b\right)^2, \qquad (2)$$

from which we have a model to predict the vehicle count y for an arbitrary image having x white pixels as $y = a^*x + b^*$.

2-3. Summary

In summary, here is how our vehicle counting algorithm works. In the training phase, for each Web camera, the prediction model is identified:

- Input: training data containing N images as $\{ (y^{(n)}, z_1^{(n)}, \ldots, z_M^{(n)}) \mid n = 1, 2, .., N \}$, where $y^{(n)}$ is the vehicle count and $z_i^{(n)}$ is the normalized pixel value of the i-th pixel.
- Output: Optimal binarization threshold k^*, predictive model parameters a^* and b^*.
- Algorithm:
  1. Find the optimum threshold k^* by solving Eq. (1).
  2. Binarize the N images to get $\{ (x^{(n)}, y^{(n)}) \mid n = 1, \ldots, N\}$, where $x^{(n)}$ is the number of white pixels of the n-th image. Note that this depends on k^*.
  3. Solve Eq. (2) for the predictive model $y = a^*x + b^*$.

In the prediction phase, the vehicle count is estimated based on the model:

- Input: A median-normalized image. Parameters k^*, a^*, and b^*.

- Output: Vehicle count y.
- Algorithm:
  1. Binarize the image using k^* to get the number of white pixels x.
  2. Compute y using $y = a^*x + b^*$.

Note that this model does not use any handcrafted features of the images. After the ROI selection, all of the steps rely only on simple calculation without *any* hand-tuning. This is in sharp contract to feature-based approaches, and clearly the best approach for VLQI. While we discussed only vehicle counting, we can extend our approach to compute other metrics such as the average velocity or the congestion level. These details appear in another paper [7] due to the space limitations. We will use the term *flow* in a general way to symbolically represent such quantities related to traffic volume.

3. NETWORK INFERANCE FOR TRAFFIC FLOW ESTIMATION

Using the algorithm in Section 2, we can estimate the vehicle flow at any instant. However, this is only for the locations where Web-cameras exist. Since the number of cameras is always much smaller than the number of links in the road network, we need some technology for *extrapolation* to monitor and manage the traffic over an entire city. That was our third challenge as mentioned in the Introduction. Our goal is to estimate the traffic flow in an arbitrary link of a network, given the observed traffic flows on a limited number of the links, as illustrated in Fig. 4. Our problem resembles network tomography [4][8] and link-cost prediction [9]. However, unlike network tomography, we need to infer all of the link traffic instead of source-destination demands, and, unlike link-cost prediction, our inputs are stationary observations instead of trajectories.

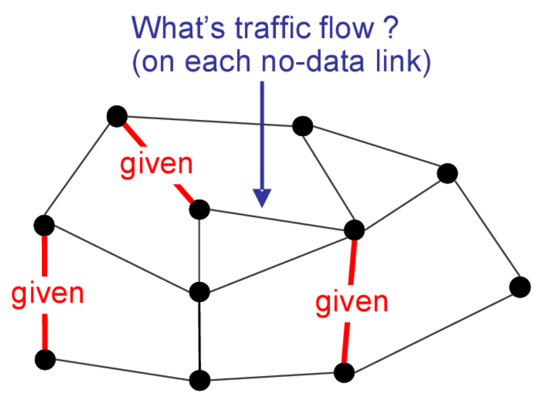

Fig. 4. Network flow inference problem.

3-1. Inverse Markov chain problem

We formalize this problem as a form of *inverse Markov chain problem*: Find the Markov transition probability $p(i \mid j)$ from an arbitrary link j to a link i, given a stationary distribution over the links, $d(i)$, i =1, .., L. Here $d(i)$ denotes the stationary state probability on the i-th link, and L denotes the number of links in the network. The basic assumption to relate the distribution to observed flows is

$$y(i) = cd(i), \tag{3}$$

where $y(i)$ denotes the observed traffic flow 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 d satisfy

$$d(i) = \sum_{j=1}^{L} p(i \mid j) d(j) , \tag{4}$$

which is also the definition of the stationary state probability. In matrix form, this equation is written as $\mathbf{Pd} = \mathbf{d}$ in the obvious notation.

Here is the high-level procedure of the traffic flow estimation problem. Starting from Eq. (3), which holds only at the links having observed data, we solve the inverse Markov chain problem to get $\mathbf{P}$ for arbitrary pairs of links. Then we re-compute $\mathbf{d}$ using Eq. (4), which is done though eigendecomposition of the probability matrix $\mathbf{P}$, to recover the flows at arbitrary links with and without observed data.

3-2. Parameterizing the transition model

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

$$p(i \mid j) = (1-\gamma) q(i \mid j; \mathbf{u}) + \gamma \cdot r(i; \mathbf{v}) , \tag{5}$$

where $0 < \gamma < 1$ is called the restart probability (assumed to be a fixed parameter), and $\boldsymbol{u}$ and $\boldsymbol{v}$ 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 [10], we first generate the starting locations and then generates finite-length paths according to a given transition rule.

For r and q, we use these particular forms:

$$\begin{aligned} q(i \mid j; \mathbf{u}) &\propto I(i \sim j) \exp\{u_{i,j} + u_0 \cos(i \mid j) + u_1 h[N_{\mathrm{L}}(i), \mathrm{type}(i)]\} \\ r(i; \mathbf{v}) &\propto \exp(v_i) \end{aligned} , \tag{6}$$

where $I(\cdot)$ is the indicator function as previously defined, and $i \sim j$ represents a connection from the j-th to the i-th links. The function $h[\ ,\ ]$ is the weighting factor depending on the number of lanes $N_{\mathrm{L}}(i)$ and the road type of the i-th link. Based on the attributes available in a digital road map, we defined 12 road types, as explained in Section 4. Also, $\cos(i \mid j)$ is the cosine between the i-th and j-th links. This model naturally reflects shared knowledge about the traffic. For example, if the j-th link points in the opposite direction as the i-th link, the transition probability between them should be down-weighted. If they point in the same direction, then that transition should occur more often.

Combining Eqs. (4), (5) and (6), we obtain the stationary distribution $d(i)$ as a function of the model parameters $\boldsymbol{u} = [u_0, u_1, u_{1,1}, \ldots, u_{L,L}]^{\mathrm{T}}$ and $\boldsymbol{v} = [v_1, \ldots, v_L]^{\mathrm{T}}$, where T denotes the transpose and $u_{i,j}$'s for unconnected pairs are omitted. Optimal model parameters are those that minimize the discrepancies between the left and right hand sides of Eq. (3). The key question is how to measure and minimize the discrepancy, which will be studied in the next subsection.

3-3. Designing the objective function

The starting point of our formulation is Eq. (3), which associates the model with observables at the links having cameras. To measure the discrepancy between *d*(*j*) and *cy*(*j*), one natural choice may be $\ln[d(i)/cy(i)]$, which is suggested by the Kullback-Leibler (KL) divergence. The KL divergence can be interpreted as the expectation of entropy loss in modeling the true distribution. Thus the term $\ln[d(i)/cy(i)]$ 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 big loss at a particular link, it should be dissipated through the transition process. With this intuitive picture in mind, we define a loss function to minimize both the information loss and the non-uniformity of the loss. Due to space limitations, however, we leave the details to a companion paper [12]. Using the loss function $L(\boldsymbol{u}, \boldsymbol{v})$, which depends on $\boldsymbol{u}$ and $\boldsymbol{v}$ through $d(i)$, the objective function to be minimized is given by

$$\Omega(\mathbf{u},\mathbf{v}) \equiv L(\mathbf{u},\mathbf{v}) + l_1\left(|u_0| + |u_1| + \sum_{i\sim j}|u_{i,j}| + \sum_{l=1}^{L}|v_l|\right) + l_2\left(u_0{}^2 + u_1{}^2 + \sum_{i\sim j}u_{i,j}{}^2 + \sum_{l=1}^{L}v_l{}^2\right), \quad (9)$$

where the second and the third terms on the right hand side are L_1 and L_2 regularizers, respectively. Although the model is highly redundant and ill-posed, the regularization terms allow giving a sensible solution. The parameters l_1, l_2 control the strength of regularization, and are determined through cross validation.

3-4. Solving the optimization problem

To minimize $\Omega(\mathbf{u},\mathbf{v})$, we employ the gradient method. We developed an efficient algorithm based on the notion of natural gradient [11], but details are omitted due to space limitations. For the details of the gradient method and others, see the companion paper [12].

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

$$c^* = \arg\min_c \sum_{j\in C}\left[y(j) - cd(j;\mathbf{u}^*,\mathbf{v}^*)\right]^2 . \quad (10)$$

Finally, the network inference algorithm is summarized as follows:

- Input: Observed traffic flows at a limited number of links $\{y(i)\}$.
- Output: $\boldsymbol{u}^*$, $\boldsymbol{v}^*$, and c^* to compute the predicted flow as $c^* d(i\,;\boldsymbol{u}^*, \boldsymbol{v}^*)$ at the link *i*.
- Algorithm:
  1. Initialize $\boldsymbol{u}$ and $\boldsymbol{v}$.
  2. Solve Eq. (3) to find $d(i\,;\boldsymbol{u}, \boldsymbol{v})$.
  3. Compute the gradient of $\Omega(\mathbf{u},\mathbf{v})$ to update $\boldsymbol{u}$ and $\boldsymbol{v}$.
  4. Proceed to 5 if there is no significant change, otherwise return to Step 2.
  5. Compute c^* using Eq. (10).

4. EXPERIMENTS

We prototyped our traffic monitoring system using Web cameras in Nairobi, provided by

AccessKenya.com [5]. In the downtown area of Nairobi, there are 1497 links, while only 52 links are monitored by the Web cameras (about 3.5%).

4-1. Vehicle counting

Figure 5 compares the actual and predicted numbers of vehicles. We used N=100 images for training. Among the 52 links, we picked only two locations for illustration. In the figure, the algorithm is tested for three different images with different time stamps. The original images appear on the left side, and the middle column shows the ROIs and binarization results. The rightmost column compares the number of vehicles between the actual and predicted counts (denoted by "pred."). Notice that the bottom row of images was taken at night.

From the figure, we see that our algorithm works quite well even when the image quality is too low to identify particular features of the vehicles such as windshields or headlights. Our method is robust to the low resolution and occlusions as well as the variation in luminance. For the nighttime images at the bottom row, in spite of the significant change in the luminance, we see that our method including the median normalization scheme works extremely well.

4-2. Network inference

We evaluated the accuracy of the algorithm in comparison to a standard method. For parameters, we fixed $l_1 = 1$, and initialized $\{v_i, u_{i,j}\}$ to 0, while other parameters in Eq. (6) are initialized to unity. For the road-type weight h, we used $\ln[N_L(i) + 1]$ multiplied by the values listed in Table 1. For l_2 and g, we used cross-validation to choose the values. Thanks to the L_1 regularizer, more than 70% of $\{v_i, u_{i,j}\}$ became zero after optimization.

The method compared is Nadaraya-Watson kernel density estimation, 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 based on the number of hops in the road network [13]. Based on a leave-one-out evaluation, we confirmed our approach is about two times better than the standard method in terms of the relative root mean absolute error.

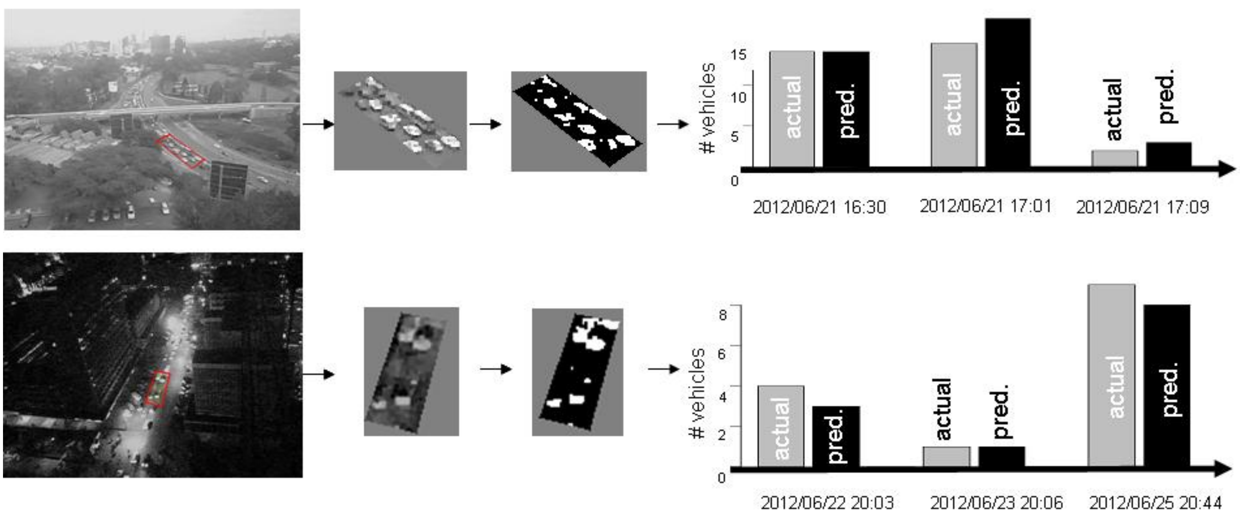

Fig. 5. Results of vehicle counting.

Table 1. Road type weight for Eq. (6).

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

Figure 6 shows an example for our results of network inference. We initially measured the vehicle flow at 36 locations [5] in Nairobi using the VLQI analysis algorithm presented in Section 2, and then estimated the flow on every link of the road network. In the right-hand panel of Fig. 6, the red and yellow roads are most congested, while the traffic on the blue roads is flowing smoothly. The congested roads from our analysis are consistent with those from a local traffic survey report [14].

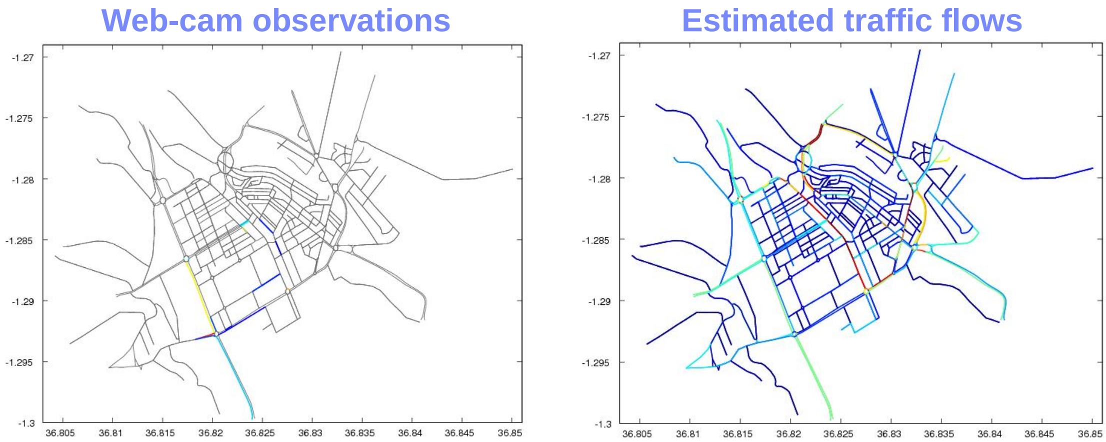

Fig. 6. Results of network flow estimation for Nairobi.

5. CONCLUSION

We have proposed a new approach to ITS. 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 traffic flow from very low-quality Web camera images, while the network flow estimation module features a traffic flow estimation algorithm for every single link, based on camera-based measurements at a limited number of links. Using real Web cameras deployed in Nairobi, Kenya, we assessed the accuracy of our approach. To the best of authors' knowledge, this is the first practical framework for monitoring an entire city's traffic without special and expensive infrastructure and time-consuming data calibrations.

REFERENCE

[1] Navi Radjou, Jaideep Prabhu, Simone Ahuja, *Jugaad Innovation*, Jossey-Bass, 2012.

[2] Kostia Robert, “Video-based traffic monitoring at day and night vehicle features detection tracking,” Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems (ITSC '09), pp.1-6, 2009.
[3] Norbert Buch, Sergio A. Velastin, and James Orwell, “A Review of Computer Vision Techniques for the Analysis of Urban Traffic,” IEEE Transactions on Intelligent Transportation Systems, 12(3), pp.920-939, 2011.
[4] Simone Santini, “Analysis of traffic flow in urban areas using Web cameras,” Proceedings of IEEE Workshop on Applications of Computer Vision, pp. 140-145, 2000.
[5] AccessKenya.com, http://traffic.accesskenya.com/.
[6] N. Otsu, “A threshold selection method from gray-level histogram,” IEEE Transactions on Systems, Man and Cybernetics, Vol.9, pp.62-66, 1979.
[7] T. Katsuki, T. Morimura, T. Idé, “Bayesian Unsupervised Vehicle-Counting,” in preparation.
[8] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, “An information-theoretic approach to traffic matrix estimation,” In Conference on Applications, technologies, architectures, and protocols for computer communications, pp. 301–312. ACM, 2003.
[9] T. Idé and M. Sugiyama, “Trajectory regression on road networks. In Proceedings of AAAI Conference on Artificial Intelligence,” pp. 203–208, 2011.
[10] T. Osogami, T. Imamichi, H. Mizuta, T. Suzumura, T. Idé, “Toward simulating entire cities with behavioral models of traffic,” IBM Journal of Research and Development, 57 (5), 2013, to appear.
[11] S. Amari. “Natural gradient works efficiently in learning,” Neural Computation, 10(2):251–276, 1998.
[12] T. Morimura, T. Osogami, T. Idé, “Solving inverse problem of Markov chain with partial observations,” in preparation.
[13] T. Idé, S. Kato, “Travel-Time Prediction using Gaussian Process Regression: A Trajectory-Based Approach,” Proceedings of 2009 SIAM International Conference on Data Mining (SDM 09), pp.1185-1196, 2009.
[14] Eric J. Gonzales et al., Multimodal transport in Nairobi, Kenya: Insights and recommendations with a macroscopic evidence-based model. Paper Number 11-3045. Transportation Research Board 90th Annual Meeting, 2011.