Accurate Speed and Density Measurement for Road Traffic in India

Rijurekha Sen[1]*, Andrew Cross[2], Aditya Vashistha[2],
Venkata N. Padmanabhan[2], Edward Cutrell[2], and William Thies[2]

[1]IIT Bombay
riju@cse.iitb.ac.in

[2]Microsoft Research India
{t-across,t-avash,padmanab,cutrell,thies}@microsoft.com

ABSTRACT

Monitoring traffic density and speed helps to better manage traffic flows and plan transportation infrastructure and policy. In this paper, we present techniques to measure traffic density and speed in unlaned traffic, prevalent in developing countries, and apply those techniques to better understand traffic patterns in Bengaluru, India. Our techniques, based on video processing of traffic, result in about 11% average error for density and speed compared to manually-observed ground truth values. Though we started with intuitive and straight-forward image processing tools, due to a myriad of non-trivial issues posed by the heterogeneous and chaotic traffic in Bengaluru, our techniques have grown to be non-obvious. We describe the techniques and their evaluation, with details of why simpler methods failed under various circumstances. We also apply our techniques to quantify the congestion during peak hours and to estimate the gains achievable by shifting a fraction of traffic to other time periods. Finally, we measure the fundamental curves of transportation engineering, relating speed vs. density and flow vs. speed, which are integral tools for policy makers.

1. INTRODUCTION

Traffic congestion leads to long and unpredictable commute times, environmental pollution and fuel waste. These negative effects are more acute in developing countries like India, where infrastructure growth is slow because of cost and bureaucratic issues. Intelligent traffic management and better access to traffic information for commuters can help alleviate congestion issues to a certain extent. Static sensors like loop detectors [2, 11, 13], video cameras [6, 7] and mobile sensors like GPS in vehicles [8, 9, 26, 31, 33] are used for traffic monitoring purposes in developed contexts. Most of these techniques, however, are only suited for lane based orderly traffic in developed countries where the penetration of GPS devices and smartphones is also sufficiently high.

Traffic in developing countries has significantly different characteristics from traffic in developed countries. High heterogeneity in vehicle sizes makes defining lanes cumbersome. On a typical road in India, all vehicles from buses to two-wheelers pack themselves together to utilize the available road infrastructure. Traditional traffic sensors have been designed to work for laned traffic, and without careful modification to sensor placement and sensor data processing, they are not suitable for monitoring unlaned traffic. We have anecdotal evidence of this shortcoming, as the image processing software from Bosch gives 55% median error on vehicle counts at the Bengaluru traffic management center [24], and hence has been replaced by manual monitoring of the video feeds to ensure accuracy.

There have been several efforts to design clean slate solutions for unlaned traffic monitoring. One approach is to detect congestion based on acoustic sensing [29], since chaotic traffic is often noisy. Another system measures traffic density (the fraction of roadway occupied by vehicles) based on variations in an RF signal between wireless sensors placed on either side of the road [28]. This system infers the length of traffic queues from an array of such sensors leading up to a traffic light. These methods are insufficient because firstly, they only yield a binary classification of traffic densities into two-states: congested or free-flowing. Secondly, both methods require significant manual overhead to train the sensors for different road characteristics [30].

There have been several initial attempts to process images and videos of chaotic traffic [17, 18, 20, 25]. Jain et al. give two algorithms [17, 18] to produce a binary classification of traffic density, one algorithm for daylight conditions and the other for nighttime. This research makes significant contributions as the techniques work on low-quality webcam images, and not traditional high-quality video feeds from traffic cameras. The algorithm for nighttime is intuitive, but the algorithm for daytime processing is more complicated, using a histogram of grayscale images to classify traffic which requires manual training; ultimately, this algorithm is similarly limited to a binary classification of traffic. There has not been an empirical evaluation of the accuracy of the algorithm yet, which leaves uncertainty about its effectiveness in detecting vehicles that match the grayscale color of the road.

Some techniques [12, 19] track vehicle features to calculate vehicle trajectories across frames and have been thoroughly evaluated in lane-based orderly traffic. Similar feature-based tracking in unlaned traffic is done by Trazer [20]. It uses Haar features to detect vehicles anywhere in a given frame and classifies each vehicle into one of five categories: heavy vehicles, medium vehicles, small four-wheelers, auto rickshaws and two-wheelers. Searching for Haar features over an entire frame needs 8-core processors to run in real time, 4 GB RAM and a 650 GB hard-drive in both online and offline computation. Also, in high traffic density, vehicle occlusion makes feature matching challenging.

The first author started work on this project during an internship at Microsoft Research India.

DEV'13, January 11–12, 2013, Bangalore, India.
Copyright 2013 ACM 978-1-4503-1856-3/13/01 ...$15.00.

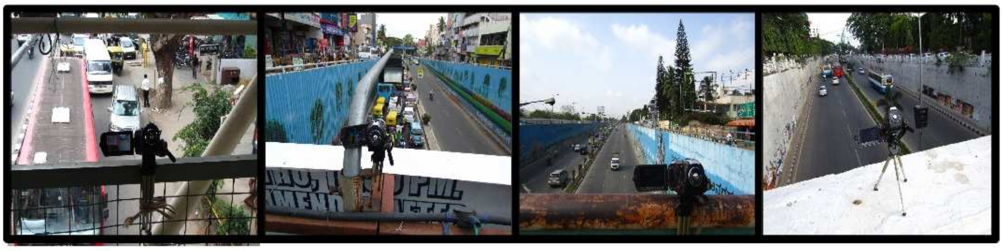


Figure 1: Our video recorder mounted at Indiranagar, Malleshwaram, Mekhri Circle and Windsor, in Bengaluru, India.

Quinn and Nakibuule [25] describe a preliminary effort to compute and group motion vectors between consecutive video frames to show position and velocity of vehicle motion. There has not yet been an evaluation comparing the velocity values to ground truth values to evaluate accuracy of the system.

Due to the limitations of the current state of the art detailed above, we present in this paper techniques to measure both traffic density and speed from video processing of chaotic traffic in the Indian city of Bengaluru. Our density measures are continuous, improving upon binary classifications of free-flow and congested and yielding the precise fraction of the road occupied. The density computations are done in real-time on dual-core processors and hence are directly usable for traffic monitoring in control rooms like Mapunity [4]. The density measures show about 11% error relative to manual ground-truth measurements and are robust across vehicles such as auto rickshaws, buses, cars, and two-wheelers. Our work can be enhanced with prior research on automated camera calibration [14], as currently we do manual calibration. Also, our density algorithm works under daylight conditions, and can be combined with prior research on night vision [10, 18].

Our speed measurement, though computationally intensive, also gives errors of about 11% relative to manually measured speed values. Along with the techniques to measure density, speed and their corresponding evaluations, a major contribution in this paper is the detailed description of the non-trivial issues we faced arising from heterogeneous vehicles and chaotic traffic conditions.

We conclude by presenting some applications of these techniques. Using 15 hours of empirical data from a certain road in Bengaluru, we calculate the morning peak hours from our density measurements. Such empirical information can incentivize commuters to shift their travel times to non-peak hours, an idea proposed by Merugu et al. [23]. Using the same data, we also present empirical plots for the fundamental curves of transportation engineering relating density vs. speed and flow vs. speed. Such empirical transportation curves can give a better understanding of traffic congestion and throughput. Though researchers have tried to characterize and model Indian traffic [21, 22, 27], efforts have either been simulation-based or limited in scale because of the manual processing overhead required, which we overcome with our automated density and speed measurement techniques.

2. EXPERIMENTAL METHODOLOGY

Our collaboration with Mapunity [4] gave us access to video data from 180 traffic cameras in Bengaluru [3]. Though these are PTZ cameras located at different road junctions that can rotate to focus on traffic flow on a particular incoming or outgoing road at that junction, the cameras' primary purpose currently is to manually observe traffic violators to penalize and fine them, especially those who jump red lights. As a result, all of the cameras are pointed at the intersection (see Figure 2), and not at any incoming or outgoing road. Because of this limitation, we were unable to use those videos for our study to measure density and speed of traffic on Bengaluru roads.

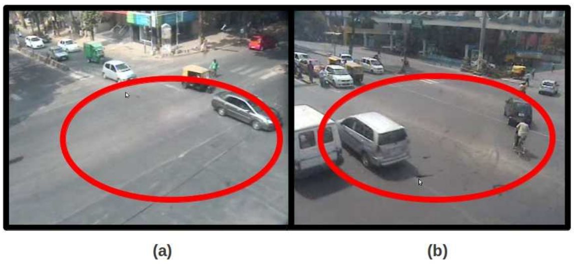


Figure 2: Police cameras at (a) Aurobindo, (b) Udipi junctions.

To capture traffic movements, we used a Canon FS100 camcorder mounted on a tripod capturing video at 25 fps from different roads around Bengaluru. The FS100 is a low-end camcorder, released in 2008, that has a resolution of 720x576; it is notable that our technique does not depend on high-end video capture. In order to get a full view of traffic, the cameras were mounted on footbridges and overpasses above the road. The pictures of the camera mounting at four different locations in Bengaluru can be seen in Figure 1. After capturing traffic flows, the videos were brought back to our lab for offline processing on an IBM R61 Thinkpad laptop. OpenCV [5], the open source video and image processing library from Intel, was used to process the videos.

3. ESTIMATING TRAFFIC DENSITY

In this section, we describe our technique for measuring traffic density. We start our discussion by describing the final algorithm, and then proceed to describe how we improve upon simpler and more intuitive methods to account for various challenges inherent to analyzing traffic in India. Finally we evaluate our method against manually computed ground truth values to evaluate the accuracy of our final algorithm.

3.1 Density Measurement Technique

To compute traffic density, we place a colored strip, either painted or taped, horizontally across the surface on the road (see Figure 3). The strip color is in stark contrast to the color of the road, such as yellow or white against traditionally gray roads. In our deployments, we used yellow-colored duct tape stuck manually to the road, which remained in place for more than two days even in spite of heavy traffic moving over it. Our camera, mounted above the road and pointed downward, captured traffic driving over the yellow tape which was later processed by our algorithm, described below. The contrasting colors help indicate the presence and movement of various vehicle types over the tape. Since many traffic-related instructions like parking restrictions are painted on the road in yellow, drivers did not pay much attention to the tape. Had we

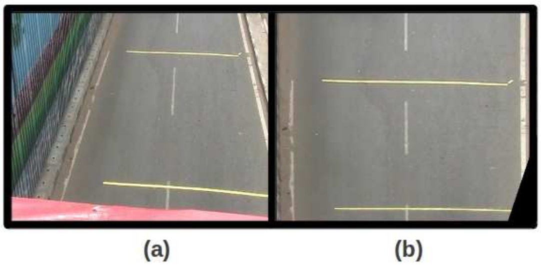


Figure 3: Road (a) before and (b) after perspective correction.

used a more out-of-place color, such as fluorescent green, that may have caused more distraction or alteration to traffic patterns.

Our basic strategy for computing density is to calculate the fraction of the tape that is obscured by vehicles on every frame. While this measurement reflects the density for only a one-dimensional strip of the frame, when averaged over time the result is proportional to the full two-dimensional frame density, assuming that vehicles cross the tape at their average speed for the frame.

To detect obfuscation of the tape, we apply two separate tests. The first test detects vehicles that have uniform coloration on their roofs, such as buses, cars, and auto rickshaws. When such a vehicle passes over the tape, it obscures the color contrast between the tape and the road. That is, without obfuscation there are neighboring pixels that have very different colors (yellow for the tape, black for the road), but with obfuscation both of these pixels are the same color (the color of the vehicle). We detect such cases by differencing the pixels immediately inside and outside the tape, at each position across the road; if the difference exceeds a threshold, we report the presence of a vehicle at the corresponding position.

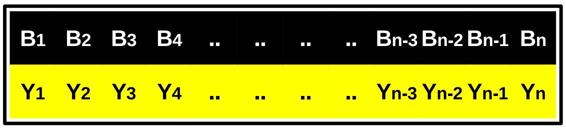
The second test detects vehicles that do not have a uniform coloration, including two-wheelers, open trucks, and the backs/sides of other vehicles. Because there is spatial variation in the vehicle's color, this implies that there is temporal variation in color as the vehicle travels over a fixed set of pixels. Thus, we detect the presence of the vehicle by detecting changes in coloration – for a fixed set of pixels overlapping the tape – between one frame and the next. Overall, a vehicle is reported if it is detected by either of the tests above.

After estimating the fraction of tape that is obscured by vehicles, we adjust this estimate to compensate for systematic bias relative to ground truth values. This adjustment is needed for two reasons. First, our algorithm still does not detect a fraction of some vehicles, leading to a systematic under-estimate of vehicle density. Second, when we extrapolate from one-dimensional density to two-dimensional density, an adjustment is needed to normalize the measurement scale. We implement the adjustment using a simple linear model: given *raw* density r, the final density $d = ar + b$, where a and b are learned using a training set of ground truth values. We describe the details of the training and testing procedures in our evaluation sections.

Our algorithm is formalized with pseudocode in Figure 4. On line 2, as the algorithm processes each frame of the video, it performs a perspective correction to make the resulting frame look more similar to one that is taken vertically from above (see Figure 3). On line 3, the algorithm divides the tape into discrete pieces, or *rectangles*; the overall density reported for a frame is the fraction of rectangles that are obscured by a vehicle. As depicted in Figure 5, each rectangle of the tape is paired with an adjacent rectangle of the road, in order to test the color contrast. We use the notation $\{B_i,Y_i\}$ to denote these corresponding *black* and *yellow* rectangle pairs. Line 4 applies both of the tests described earlier: (i) testing changes in color between the tape and the road, and (ii) testing changes in color from one frame to the next. Line 5 computes the raw frame density in terms of the fraction of rectangles that are obscured. Line 6 computes the corrected frame density using a linear model, in which the parameters are gleaned from training data.

```
1: function COMPUTE-DENSITY
2:     Correct perspective error of camera for frame $N$ (shown
           in Figure 3).
3:     Divide the rectangular yellow tape and a parallel black
           rectangle of the road, adjacent to the tape, into vertical
           rectangle pairs $\{B_i,Y_i\}$ (shown in Figure 5).
4:     For each $\{B_i, Y_i\}$ pair, consider the pair occupied if either:
           (i) the RGB difference between $B_i$ and $Y_i$ is below a
                   threshold $C$, or
           (ii) the RGB difference between $Y_i$ in frame $N$ and $Y_i$
                   in frame $N - 1$ is more than a threshold $T$.
5:     Compute *raw* density $r$ as ratio of occupied rectangle pairs
           to total rectangle pairs.
6:     Compute *corrected* density $d = ar + b$, where $a$ and $b$ are
           learned from training data.
7: end function
```

Figure 4: Algorithm to compute density for frame N.



B1	B2	B3	B4	..	..	..	..	Bn-3	Bn-2	Bn-1	Bn
Y1	Y2	Y3	Y4	..	..	..	..	Yn-3	Yn-2	Yn-1	Yn

Figure 5: Vertical pairs of black and yellow rectangles.

3.2 How We Arrived at this Technique

Though the algorithm described above is complex and non-intuitive, the complexities are necessary to accurately measure traffic density due to several challenges for image processing. Here we describe the simpler methods that we tried, the obstacles we encountered, and the solutions we came up with to arrive at the final algorithm.

Background subtraction does not detect buses

Initially, we tried background subtraction, a well-known image processing tool. In this method, the foreground pixels in frame N are calculated by subtracting a background frame from it. The density of frame N is calculated by the ratio of foreground pixels to total pixels in the frame. The background frame was manually selected as a frame containing no vehicles to serve as the template, where any pixel differences indicate a vehicle. This simple intuitive method gave disastrous results because of a peculiar characteristic of buses in Bengaluru.

Bengaluru buses have a gray cover on their roofs, probably as a protection from heat and rain. Online image searches revealed similar characteristics of buses in other Indian cities. This gray color is almost the same color as the road, and therefore using background subtraction does not detect a vehicle. In fact, when two buses stand side by side occupying the entire road as shown in Figure 6, background subtraction measures no density (Figure 7).

Simple yellow-tape analysis is too sensitive to lighting

In order to handle the issue of gray bus tops, we introduced the novel idea of using a high-contrast strip of tape on the road. As described previously, we defined vertical rectangle pairs $\{B_i,Y_i\}$ and computed density as the ratio of occupied rectangle pairs to total

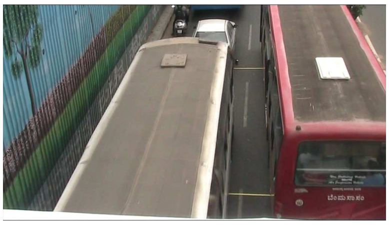


Figure 6: Video excerpt with two buses, a challenging case for density calculation.

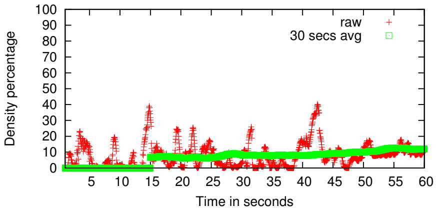


Figure 7: Background subtraction does not detect the buses in video excerpt.

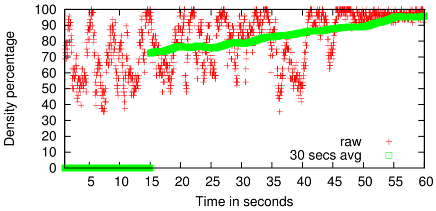


Figure 8: Yellow tape analysis (simple algorithm, not our best) detects the buses.

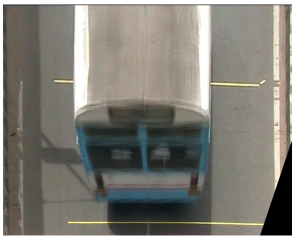


Figure 9: A bus with its shadow. Simple techniques are sensitive to lighting conditions and may count the shadow as part of the vehicle; our technique reports the correct density.

rectangle pairs. Our initial approach to detecting "occupied" pairs was very simple. Building on the idea of background subtraction, we manually selected a background frame without any vehicles, containing only the empty road with the tape. For a frame N, if the RGB color values of pixels in either B_i *or* Y_i differed from corresponding pixels in the background frame by more than a threshold T, the pair $\{B_i,Y_i\}$ was considered occupied in that frame, indicating the presence of a vehicle. This approach detects gray buses due to changes observed to the yellow rectangle Y_i. It detects auto rickshaws (which are yellow) due to changes to the black rectangle B_i. The algorithm also detects vehicles of other colors, as they change the appearance of both rectangles.

This simple solution handled the issue of gray bus roofs, as seen from Figure 8 where density in a frame with two buses is measured accurately. However, the shortcoming of this method is that it is not robust to variable lighting conditions, in particular shadows. Surprisingly, the shadows of vehicles caused the pixel colors to change so much that no value of the threshold T was able to filter the shadows and detect the vehicles accurately. Depending on the angle of light, sometimes density was measured to be higher than the actual density because the shadow made the vehicles look bigger than they actually were. An example of a vehicle with shadow is shown in Figure 9. Also, for both this technique and for background subtraction, the manual selection of the background frame was tricky. Lighting conditions changed over time, which meant an empty frame under different lighting conditions could exhibit color differences relative to the background frame, leading to non-zero density measurement.

Our final algorithm is robust to these lighting variations by examining only the difference in color between black and yellow rectangles, as opposed to comparing either rectangle to an absolute threshold. For example, in Figure 9, we do not mistake the shadow as being part of the vehicle.

3.3 Evaluation

To evaluate our technique, we use two levels of analysis. The first considers individual cars and compares the fraction of the tape obscured relative to manually-measured values. The second considers a longer period of general traffic and compares to a different ground truth metric: the fraction of the entire frame that is occupied by vehicles.

Individual vehicles

We consider ten vehicles of each type (auto rickshaws, buses, cars, and two-wheelers), for a total of 40 vehicles. To simplify this analysis, we restrict our attention to vehicles that pass through the field of view alone (i.e., there is no other vehicle visible in frames where they are visible).

Ground truth: Our ground truth in this comparison is the fraction of the tape that is obscured on each frame, as collected by manual measurements. To collect this data, we examined each frame of the video in which a vehicle overlapped the tape (488 frames total) and measured the number of pixels obscured (by any part of the vehicle) using a screen capture tool.

Automated algorithm: The automated algorithm worked as described in the prior section, and was applied to every frame in which the vehicle appeared (including frames in which the vehicle did not overlap the tape). Where indicated below, we aggregated the per-frame density by vehicle type to obtain an average density for each vehicle in the frame. For illustrative purposes, we did not apply any correction (line 6 of the algorithm) except in computing the relative error rates; in this case, we trained the model on the per-vehicle ground truth densities using least-squares linear regression and leave-one-out cross validation.

Results: First we describe the raw density calculations (from line 5 of the algorithm), in order to understand the accuracy of the basic technique. Then we use the corrected density estimates in order to compute the error rate of our technique.

Figure 10 illustrates the raw fraction of the tape obscured on a frame-by-frame basis, using both automated and manual measurements. We observe a good correlation ($R^2 = 0.81$) between the calculated values and the ground truth. Errors originate from two sources. First, some parts of vehicles are not detected as obscuring the tape, which results in points that fall below the trend line in Figure 10. Second, a handful of shadows are detected as being part of the vehicle, which results in points along the y axis in Figure 10.

To estimate the impact of these frame-level errors on the aggregate density measurement, we average raw density values across frames corresponding to a single vehicle. Results of this per-vehicle density comparison appear in Figure 11. At the granularity of vehicles, we observe a much higher correlation ($R^2 = 0.998$) between calculated and ground truth values.

The error rates[1] of our final (corrected) density calculations are shown in Table 1. Overall, our estimates show an average error rate

[1] Throughout this paper, the error rate e for a single calculation is defined as $e = |(c - t)/t|$, where c represents the calculated value and t represents the true value. When averaging the error rate across multiple predictions, we ignore the few cases where $t = 0$.

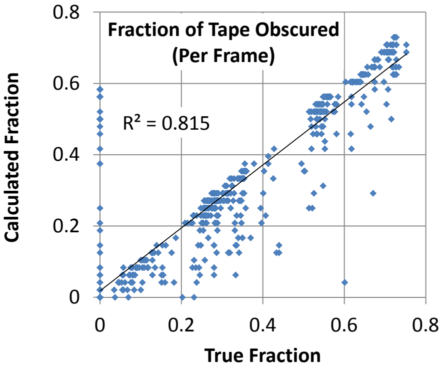


Figure 10: Comparison of frame-level density calculations with ground-truth values. Density estimates represent raw values (line 5 of algorithm), prior to correction.

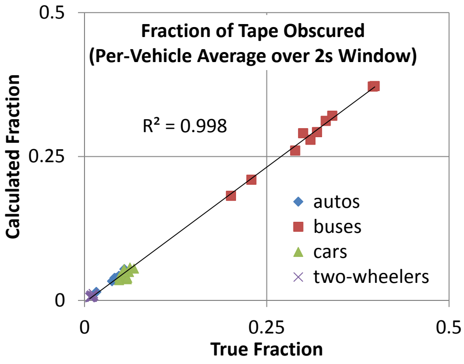


Figure 11: Comparison of vehicle-level density calculations with ground-truth values. Density estimates represent raw values (line 5 of algorithm), prior to correction.

vehicle type	average error	median error
auto rickshaws	9.3%	5.4%
bus	1.8%	1.2%
car	9.6%	11.4%
two-wheelers	13.8%	15.0%
all vehicles	**10.9%**	**5.0%**

Table 1: Density error rates per vehicle type. Densities represent corrected values, separately trained and tested for each row of the table, using the vehicle type(s) in the first column.

of 10.9%, and a median error rate of 5.0%. Error rates are lowest for buses (1-2%) and highest for two-wheelers (14-15%) due to the size of those vehicles; for smaller vehicles, a fixed amount of error results in larger error rate.

General traffic

In this section we expand the evaluation to a more realistic scenario, in two respects. First, we consider general traffic, in which many vehicles may be within the field of view at the same time. Second, we estimate the density for the full two-dimensional frame, and compare to corresponding ground truth values, rather than restricting our attention to the fraction of tape that is occupied.

Our analysis utilizes three video segments from Malleshwaram. In total, the videos last 130 seconds and include 122 vehicles. Congestion varied across the segments, but was generally medium to light. We did not evaluate bumper-to-bumper, stand-still traffic due to the overhead of manual data collection; stand-still traffic requires a large window of samples in order for the one-dimensional density measured at the tape to match the two-dimensional density of the frame.

Ground truth: To ascertain the ground-truth density, we manually analyzed each frame of the videos (about 3250 frames total). After correcting the videos for perspective (as per Figure 3), we measured the width and length of each vehicle, as well as the frames in which points of interest on the vehicle (front and back edges) both entered and exited the frame. Assuming constant vehicle motion, this information is sufficient to calculate the fraction of each vehicle that is visible on each frame.

One limitation of our ground-truth data is due to the camera perspective. Because we were not looking straight down on traffic, our videos include a view of the back side of vehicles (even following perspective correction). Due to the difficulty of automatically distinguishing between the top and the back of vehicles, for the sake of this comparison we counted both the top and back of vehicles as part of the ground truth density. This introduces bias in two ways: first, taller vehicles are reported to consume more space than they actually do, and second, vehicles appear to grow as they progress across the frame. For simplicity, we represent the vehicle size as constant across frames; this size is calculated as the average of the size upon entering the frame and the size upon exiting the frame.

Automated algorithm: For automatic calculation of density, we started with the algorithm described previously to estimate the raw fraction of tape that was obscured by vehicles on each frame. In order to extrapolate to the density of a two-dimensional frame, we simply average the one-dimensional densities over a window of values. We also make a small adjustment to compensate for the fact that the tape was at the bottom of the frame: to estimate the full frame density at frame N, we consider the tape values averaged over a window centered at $N-7$. Also, we correct the final estimates by training against ground truth values for full-frame densities (using linear regression and leave-one-out cross validation).

Results: Results of the comparison are illustrated in Figure 12. Our technique performs well, eliciting a good correlation between predicted and true densities ($R^2 = 0.87$). The average and median error rates are 18% and 15%, respectively. This comparison utilizes a window size of 6s, which is about the largest window that is meaningful using our limited ground truth data.

Because our algorithm relies on averaging to extend the measured one-dimensional density to the full two-dimensional frame, our results improve considerably with increasing window sizes. This trend is illustrated in Figure 13, where average error rates decrease roughly ten-fold as the window grows from 1s to 6s. In the applications described later, we use a window size of 30s, which should yield even better accuracies for our technique.

4. ESTIMATING TRAFFIC SPEED

In this section we describe our speed estimation technique. Similar to our explanation of the density estimation technique in the previous section, we start by presenting the final algorithm followed by a description of the inadequacies of simpler methods that prompted our improvements culminating in the design of the final algorithm. We conclude by evaluating the accuracy of our speed estimates relative to the manually observed ground truth.

4.1 Speed Estimation Technique

The algorithm to compute traffic speed for a frame N is given in Figure 14. The goal is to find the displacement or offset between pixels of two consecutive frames that maximizes the similarity between pixels of those two frames. In other words, if all pixels in

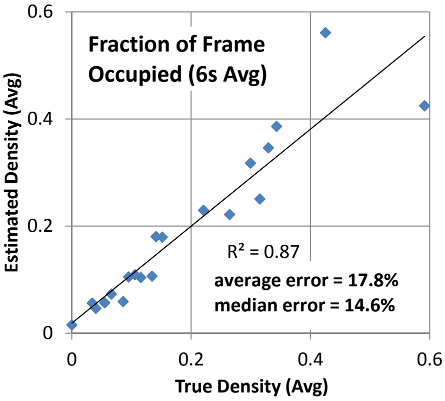


Figure 12: Correlation between calculated and true density values for 6s windows of traffic footage.

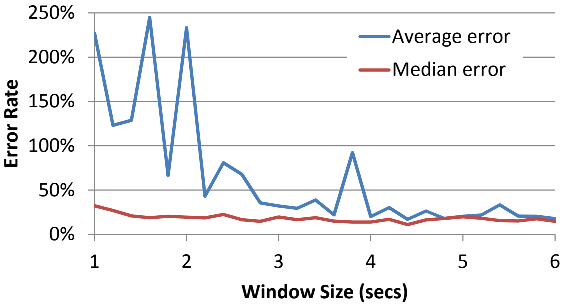


Figure 13: Density errors as a function of window size.

frame $N-1$ moved by that offset, then they would best match the pixels in frame N.

We first correct perspective errors in both frames $N-1$ and N. Then we consider pixels *PixelSet* in frame N that have changed by more than a threshold from the corresponding pixels in frame $N-1$. This reduces the search space by ignoring stationary pixels, where no new vehicles have entered and no old vehicles have left. This also prevents mistaking the yellow tape for a car with speed zero.

Then we consider pixels *PixelSet'* in frame $N-1$, which contain pixels within a search window distance from *PixelSet*. This search window is chosen to reduce the search space to make computation faster while still considering pixels at far enough distance to handle high speed motion. With our 25 fps 720 by 576 pixels video, we empirically found that a search window of 100 pixels is more than sufficient to match vehicles moving at high speed.

The algorithm then finds the (x,y) offset between pixels *PixelSet* in frame N and pixels *PixelSet'* in frame $N-1$, that minimizes the total RGB difference over all pixels in *PixelSet*. This offset represents the *raw* estimate of speed between frames $N-1$ and N. Analogously to our density algorithm, we adjust the raw estimate into a *corrected* value using a linear model based on training data, which is returned from the algorithm.

4.2 How We Arrived at this Technique

Like density, our speed measurement technique also evolved from simpler methods. Researchers previously have used motion vectors to estimate vehicle motion [16, 25, 32]. As the embedded motion vector values in MPEG2 videos are computed over a small search space, for improved accuracy, we tried to compute motion vectors between consecutive frames using OpenCV functions. The magnitude of the motion vectors computed for each pixel in a frame, was considered as the speed for that pixel.

```
function COMPUTE-DIFF(frame 1, frame 2, pixelSet)
    for each pixel $p$ in pixelSet do
        RGBdiff = RGBdiff + difference in RGB between $p$ in
              frame 1 and $p$ in frame 2
    end for
    return RGBdiff
end function
function COMPUTE-SPEED(frame $N-1$, frame $N$)
    Correct perspective error of camera for both frames $N-1$
          and $N$ (shown in Figure 3).
    Let $S$ be the set of pixels $p$ s.t. RGB difference
          between $p$ in frame $N-1$ and $p$ in frame $N$ is greater
          than threshold $T$
    Select the offset $(i,j)$ within a search window, that
          minimizes COMPUTE-DIFF(frame $N-1$ shifted by
          $(i,j)$, frame $N$, S). The raw speed is set to $j$.
    Compute the corrected speed $s = aj + b$, where $a$ and $b$
          are learned from training data.
    return s
end function
```

Figure 14: Algorithm to compute speed for frame N.

Though moving vehicles were detected accurately using the OpenCV motion vectors, when we looked at actual magnitudes of speeds, and compared them to manually measured pixel movements between frames, we found the correlation to be extremely low (0.07 for one ground truth dataset). One challenge was that the motion vectors were being computed independently for small blocks of pixels; thus, pixels within a homogeneous region (such as roofs of cars) might find a good match at a distance or direction different than the velocity of traffic. Our algorithm can be understood as a variation of MPEG2 motion estimation, but rather than considering small macroblocks, we consider a large set of pixels (those that have changed between frames), thereby averting this problem.

It may be that techniques that track flow of features [12, 19, 20, 25] can give equal or better speed measurements than we present here. However, we are unaware of any published evaluation of feature-based techniques in unlaned traffic. The simplicity of our technique may also make it more accessible to non-experts.

4.3 Evaluation

Similarly to the evaluation of density, the evaluation of our speed algorithm proceeds in two steps: first for individual vehicles, and then for general traffic.

Individual vehicles

We examine the speeds of 40 vehicles (10 of each type), using the same data set as described in the density section.

Ground truth: We assume that each vehicle travels at a constant rate, and calculate its speed based on the number of pixels traveled in a given number of frames. We consider a window of frames that is as large as possible while keeping the vehicle in view, and measure the number of pixels traveled during that time. To minimize distortion due to perspective, we track a point on each vehicle that is as close to the ground as possible (e.g., the lower-back edge, near the rear tires).

One limitation of this data is that the assumption of constant vehicle speed is sometimes violated, as vehicles accelerate or decelerate in response to upcoming road conditions.

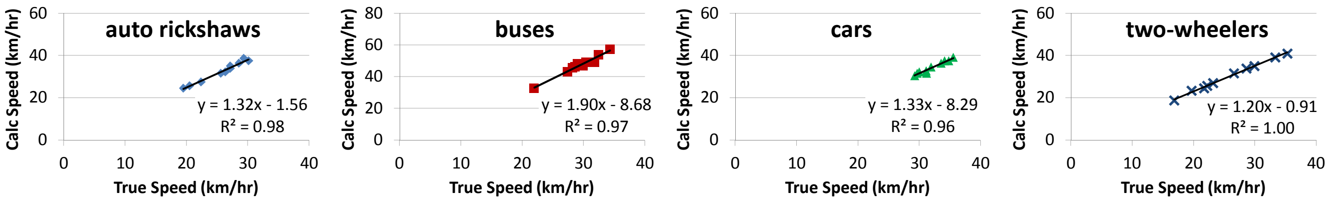


Figure 15: Correlation between calculated and true speed values for individual vehicle types.

Automatic algorithm: To condense the algorithm's per-frame estimates into a single estimate for each vehicle, we use the median value out of the frames in which the vehicle is present.

Results: The raw speeds (line 10 of the algorithm) are shown for each vehicle type in Figure 15 and for all vehicle types combined in Figure 16. The final error rates of corrected speed values are summarized in Table 2.

These results reveal an interesting and important trend: the raw speed values correlate better with ground truth for individual vehicle types than they do for all vehicles combined. For individual vehicle types, the algorithm leads to correlations of at least $R^2 = 0.96$. However, for all vehicles combined, the correlations is low ($R^2 = 0.48$).

The cause of this trend is that different vehicles have different heights, and taller vehicles appear faster to our algorithm because they are closer to the camera. For example, in Figure 16, the speed calculated for buses is high relative to all of the other vehicles, because buses are taller and closer to the camera.

Thus, in order for our technique to accurately measure the speed of individual vehicles, it seems necessary to detect the vehicle's height. Perhaps this could be done with a stereo camera that calculates the depth of each pixel, or an auxiliary device (such as a rangefinder) that calculates distance to a target. Another approach would be to utilize computer vision to detect the vehicle type, and consequently infer the approximate height.

General traffic

Ground truth: We utilize the same data set as described in the density evaluation. Our calculation of speed values is similar to the single-vehicle case. However, in order to enable coding of more ground-truth data with limited human resources, we utilized a more approximate measurement. Instead of measuring the exact number of pixels traveled by cars, we measured the number of frames needed to traverse the full field of view (576 pixels).

If a frame contains more than one vehicle, then we calculate the overall speed for the frame as the weighted average of those vehicle speeds, where the weights correspond to the ground-truth densities for the vehicles. For example, if a bus and two-wheeler are traveling at different rates in the same frame, then the overall speed will give more weight to the speed of the bus, since it is a larger vehicle. In computing the speed for a frame, we consider all vehicles that have any part visible (including their back) within that frame.

Our coding has limited precision, as most vehicles require about 20 frames to cross the field of view. This implies an inherent imprecision of $\pm 2.5\%$ for all vehicles, which is significant relative to our algorithm's error rate. In fact, one could argue that our algorithm is, by construction, a better estimate of the true speed. Nonetheless, we explore the correlation between both measurement techniques as a rough validation of our technique.

Automated algorithm: We start by calculating the raw speed values as described on line 10 of the algorithm. In order to discard a handful of outlier values, we apply a five-point sliding median across frames. Then, to compute the average speed over a window, we consider all non-zero values within that window. If there are no cars within a window, then we report the speed as zero. Finally, we adjust the windowed speed values using a linear transformation, analogous to line 11 of the algorithm. We learn the parameters by training on windows of ground-truth speed values, using least squares linear regression and leave-one-out cross-validation.

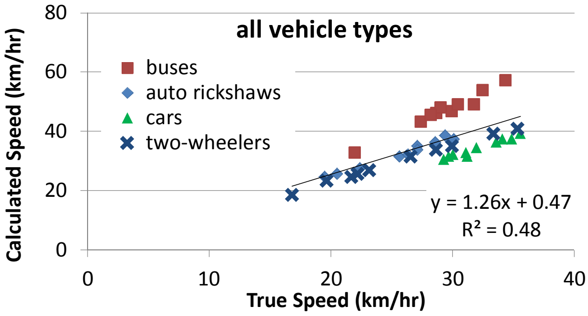


Figure 16: Correlation between calculated and true speed values for all vehicle types.

vehicle type	average error	median error
auto rickshaws	2.4%	2.4%
bus	1.7%	1.3%
car	1.3%	1.0%
two-wheelers	1.6%	1.7%
all vehicles	**11.4%**	**11.2%**

Table 2: Speed error rates per vehicle type. Speeds represent corrected values, separately trained and tested for each row of the table, using the vehicle type(s) in the first column.

Results: Illustrated in Figure 17 are the results for a 6s window, the largest that is meaningfully supported by our ground truth data set. The algorithm performs accurately, with a high correlation to ground truth ($R^2 = 0.94$), a low average error (7.7%), and a low median error (5.6%). Figure 18 shows that the error rate decreases for larger window sizes, suggesting that the 30s window used in our applications section is even more accurate than reported here. These results are better than for the individual vehicle case because there is a non-uniform distribution of vehicle types (in particular, fewer buses) in real traffic.

5. APPLICATIONS

Having shown the accuracies of our density and speed measurement techniques, we now present some possible applications. The first application is to present current traffic conditions to commuters to potentially affect their choice of travel time. The second application is to compute the relationships between different traffic parameters like density, speed and flow.

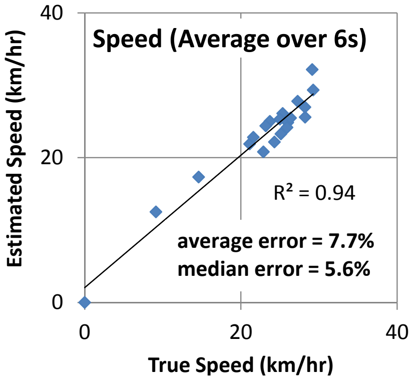


Figure 17: Correlation between calculated and true speed values for 6s windows of traffic footage.

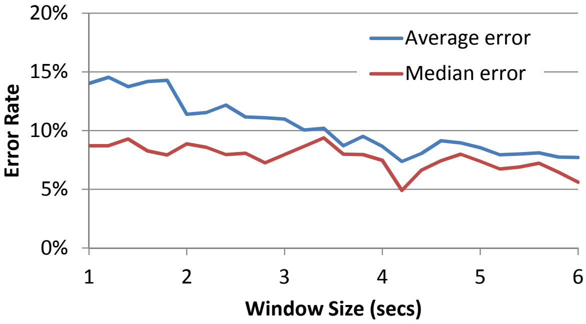


Figure 18: Error rates for speed calculations as a function of window size.

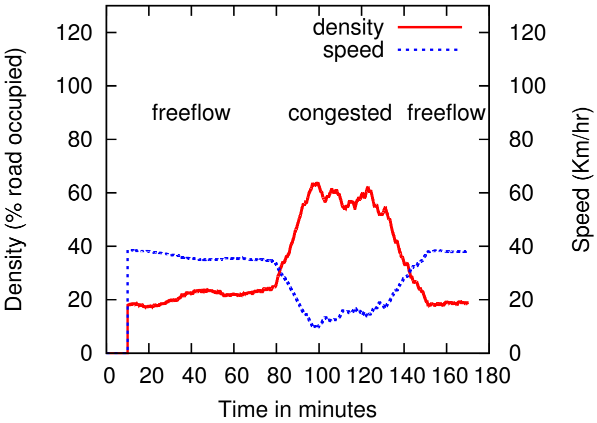


Figure 19: Traffic density and speed (20-minute moving average) from 8:15am to 11:15am on July 10.

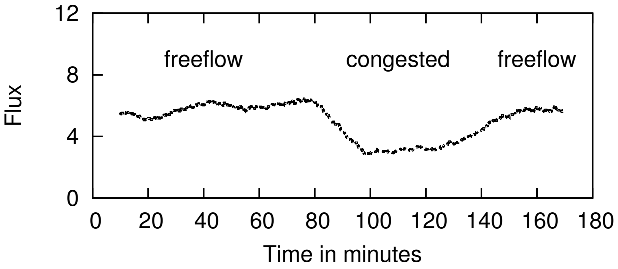


Figure 20: Traffic flux (20-minute moving average) from 8:15am to 11:15am on July 10.

5.1 Application 1: Shifting Travel Times

The notion of using alternative routes to reduce travel time is a well-known idea. In cities like Bengaluru, however, the lack of alternative routes renders the strategy ineffective. Instead of choosing different routes, Merugu et al. [23] proposed that commuters could choose different travel times to reduce commute times. To further the idea, this work also studied the concept of using economic incentives to encourage employees of a certain company to come to their office earlier than their normal schedule.

To understand the attitude of commuters towards the idea of shifting their travel times, we conducted a small survey. While vehicles waited in a long queue at a red signal on Kasturba road in Bengaluru, we polled drivers in stationary private vehicles about their destination, their choice of travel time and whether they could have traveled at a different time to avoid traffic. We conducted the survey for 4.5 hours over 3 days and questioned 20 people, spending a minimum of 6 minutes with each person. Half of the people answered that it was not possible for them to shift their travel time because of several constraints at home or work, while the remaining 10 people said that for them it was possible to drive at a different time. However, even those who were free to shift their travel time were pessimistic about the effect of doing so. They felt that *traffic conditions would have been bad even at that different time*. We concluded that there is a gap between the commuters' perceptions of traffic patterns throughout the day and the reality, which a system using our algorithm could address.

To empirically evaluate these commuters' misgivings, we collected 15 hours of video between 8:15am and 11:15am every day on July 6^{th}, 9^{th}, 10^{th}, 11^{th} and 12^{th}, 2012, on 5th Cross underpass at Malleshwaram in Bengaluru. These five dates covered every day of the work week from Monday to Friday. The 5th Cross underpass is close to a major city bus stop in a commercial area, and is a busy stretch of road on weekdays. We manually noted the road state and processed the collected videos offline to compute density and speed.

Moving averages of frame densities and speeds over 20 minutes, for 3 hours on July 10th, are shown in Figure 19. As the figure demonstrates, traffic density is low for the first hour, increases until the worst congestion is observed between about 9:55am and 10:25am, and then decreases after 10:25am. The figure also demonstrates the inverse relationship between speed and density. The pattern of congestion was observed manually as well, and was similar over all five days of data collection, though congestion was less on July 9^{th} for no observable reason.

Using Figure 19, we can revisit the question of commuter behavior. The figure shows that there are enough windows of time where people, free to travel at alternate times, could travel and encounter much less congestion for this particular road stretch. It also shows the empirical speed difference between free-flow and congested traffic on this road. Thus the pessimism that the commuters expressed in our interviews might sometimes result from a lack of information about actual traffic situation. The current manual inspection of traffic feeds in Bengaluru, in the absence of automated image processing techniques, can benefit from our techniques by offering richer data with less effort. The end result would be to provide commuters with accurate real-time or historical statistics to help them make more informed decisions about their travel times.

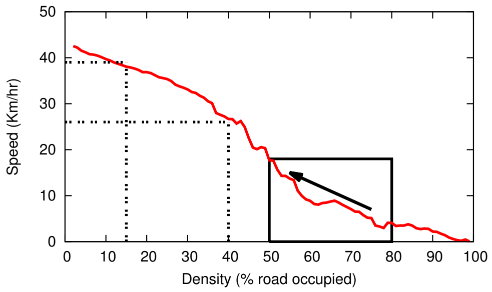


Figure 21: Speed vs. density (based on 30s moving averages).

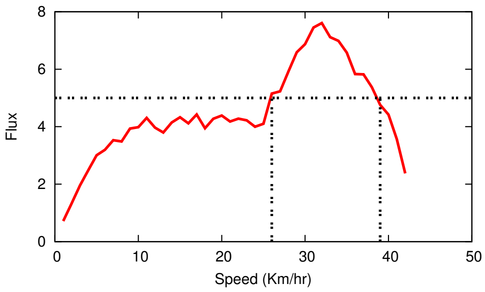


Figure 22: Flux vs. speed (based on 30s moving averages).

5.2 Application 2: Fundamental Curves

We next seek to analyze the relationship among the different traffic metrics like speed and density, and also flux, which is the product of the previous two. From an individual commuter's perspective, improving his or her own speed and hence commute time matters the most. Hence he or she may choose to travel in a period of low density. But from the perspective of traffic management, flux or throughput is also important and higher flux enables handling of more traffic at a particular intersection or road stretch. Thus higher densities, if that translates to higher flux, might be preferred.

We first plot the flux values in Figure 20 for the same three hours of data presented in Figure 19. As we see, the flux values during periods of congestion (high density) are marginally lower than in periods of free-flow (low density). Intuitively, an increase in density causes such a decrease in speed that the product of the two is lower, than at low densities with high speeds.

We next compute density, speed and flux values for the video data collected at Malleshwaram on July 10, which contains both free-flow and congested traffic. After taking 30-second moving averages of both speed and density, we average the speed values over bins of 10% density values. The resulting speed vs. density plot is shown in Figure 21. Similarly, we average the flux values over bins of 10 speed values and the resulting flux vs. speed plot is shown in Figure 22. This is the first attempt to plot these curves, also known as the fundamental curves of transportation engineering [1], using automated vision algorithms to collect empirical data on chaotic and unlaned traffic in India.

The main observations that we make from Figures 21 and 22 are as follows.

(1) The general shape of the curves in both figures matches the theoretical relationships [1] as well as empirical relationships measured for laned traffic in Atlanta, Georgia, and California [15]. Speed decreases with increasing density in Figure 21. Flux values, the product of speed and density, are low at *low speeds with high densities* on the left part of Figure 22, and also low at *high speeds with low densities* on the right. When both speed and density are high enough, flux values are high in the middle.

(2) Though the general shape of the curves matches our expectations, the exact nature of the curves might be interesting to study in comparison to similar empirical graphs. For example, the maximum traffic speeds we observe are lower than in [15]. It is not clear if this is an aspect of laned vs. unlaned traffic, or an artifact of our observation point.

(3) An increase in speed is intended by commuters and traffic management authorities, while the latter might also want to increase flux. Density is probably not a necessary metric to optimize directly. To operate at higher speeds, we want to remain as far to the left as possible in Figure 21. For higher flux, say greater than 5 as shown by a horizontal line in Figure 22, speeds between 26 and 38 km/hr are needed, as shown by vertical lines. Such speeds, as seen by dotted lines in Figure 21, will require the density to be approximately 40% or below.

(4) Though the maximum densities in Figure 21 are seen to be near 100%, such high density values occur very infrequently. Figure 23 illustrates the fraction of overall flux that appears in each of the binned density ranges for the congested period (9:45am to 10:35am) on July 10. As seen from the figure, 95% of the flux corresponds to densities of less than 80%. Let's consider a value of 75% density as typical of highly congested scenarios; this density is shown to be common in Figure 23, though it is not visible in Figure 19 due to the 20-minute moving average. Figure 21 shows that decreasing the density from 75% to 55% can double the speed. This is shown by an arrow inside the region of interest marked by a rectangle.

This observation is interesting for understanding how a shift of travel times can impact congested periods. If an additional 20% of the total roadway is cleared by time-shifting commuters, and additional vehicles do not come to take their place, then speeds for the remaining vehicles can improve considerably. This phenomenon of a small number of extra vehicles causing congestion collapse was also observed by Jain et al. [18]. The feasibility of imposing such shifts in travel times to positively impact congestion, using either incentives or congestion pricing, will be interesting to explore.

The final question we consider is as follows: *By how much would speeds improve, over a fixed period on a given day, if the total flux or throughput during that period is re-distributed with constant speed and density?* This question speaks to the "best case scenario" of traffic control, in which rush hours are eliminated and traffic speeds increase while maintaining the same volume of transportation.

To address this question, we utilize 3 hours of data from July 10. Figure 24 plots the original distribution of speeds during this time period. Observations from 30s windows are sorted by speed and aggregated according to their flux. We find the majority (about 60%) of flux during this period was at 20 to 30 km/hr. To estimate the impact of spreading out the flux, we first calculate the average flux for the time period: 5.04. Then, Figure 25 plots the distribution of speeds observed for similar values of flux (between 4.5 and 5.5). The figure shows that speeds are typically above 35 km/hr, representing a modest improvement for commuters. It remains a question for future work as to how flux can be regulated with sufficient precision to enable this benefit.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we present techniques to estimate traffic density and speed from video analysis of chaotic unlaned traffic prevalent in developing countries. Our methods are accurate to within about 11% of manually measured speed and density values. We

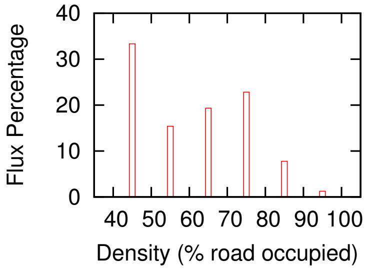


Figure 23: For 30s windows during congested periods on July 10, percentage of flux that exhibited given densities.

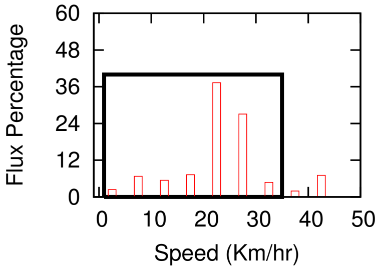


Figure 24: For 30s windows between 8:15am to 11:15am on July 10, percentage of flux that exhibited given speeds.

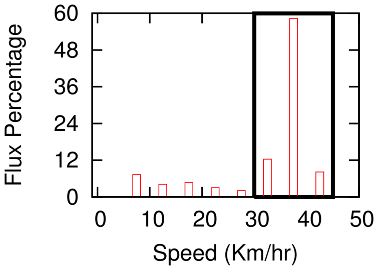


Figure 25: For 30s windows with flux between 4.5 - 5.5 on July 10, percentage of total flux that exhibited given speeds.

also present some applications for our methods in detecting peak traffic hours on a road and computing empirical relations between traffic parameters like density, speed and flow.

There are several possible extensions to our work. First, our algorithms were trained and tested in a single location and during similar times of day. Generalization across different locations and variable lighting conditions is an important task for future work. Second, the current method needs video transfer from the road to a lab for processing. One possible extension could be to implement our algorithms on TI's image processing platforms or GPUs to reduce the video communication costs. If that proves infeasible, we could potentially reduce the overhead of wired connections for video transfer and study whether wireless video transfer over 3G or 4G provides sufficient quality for video analysis. Third, we have examined temporal variations of density at a fixed location. It could be interesting to examine spatial variations of density at a fixed time, to see how different parts of a road network affect each other. Overall our methods give sufficiently high accuracy and shows good promise to be useful in real application scenarios.

7. REFERENCES

[1] http://en.wikipedia.org/wiki/Fundamental_diagram_of_traffic_flow.

[2] http://paleale.eecs.berkeley.edu/~varaiya/transp.html.

[3] http://www.btis.in/.

[4] http://www.mapunity.in/.

[5] http://www.opencv.org/.

[6] http://www.scats.com.au/index.html.

[7] http://www.visioway.com, http://www.traficon.com.

[8] R. E. Anderson, A. Poon, C. Lustig, W. Brunette, G. Borriello, and B. E. Kolko. Building a transportation information system using only GPS and basic SMS infrastructure. In *ICTD*, 2009.

[9] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson. EasyTracker: automatic transit tracking, mapping, and arrival time prediction using smartphones. In *SenSys*, 2011.

[10] Y. L. Chen, B. F. Wu, H. Y. Huang, and C. J. Fan. A real-time vision system for nighttime vehicle detection and traffic surveillance. *IEEE Transactions on Industrial Electronics*, 58(5), 2011.

[11] S. Cheung, S. Coleri, B. Dundar, S. Ganesh, C. Tan, and P. Varaiya. Traffic measurement and vehicle classification with a single magnetic sensor. Paper ucb-its-pwp-2004-7, CA Partners for Advanced Transit & Highways, 2004.

[12] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C*, 1998.

[13] B. Coifman and M. Cassidy. Vehicle reidentification and travel time measurement on congested freeways. *Transportation Research Part A: Policy and Practice*, 36(10):899–917, 2002.

[14] D. J. Dailey, F. W. Cathey, and S. Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. *IEEE Transactions on Intelligent Transportation Systems*, 1(2), 2000.

[15] W. H., L. J., C. Q., and N. D. Representing the fundamental diagram: The pursuit of mathematical elegance and empirical accuracy. In *Transportation Research Board 89th Annual Meeting*, 2010.

[16] F. Hu, H. Sahli, X. F. Dong, and J. Wang. A High Efficient System for Traffic Mean Speed Estimation from MPEG Video. In *Artificial Intelligence and Computational Intelligence*, 2009.

[17] V. Jain, A. Dhananjay, A. Sharma, and L. Subramanian. Traffic density estimation from highly noise image sources. *Transportation Research Board Annual Summit*, 2012.

[18] V. Jain, A. Sharma, and L. Subramanian. Road traffic congestion in the developing world. In *ACM DEV*, 2011.

[19] N. K. Kanhere, S. T. Birchfield, W. A. Sarasua, and T. C. Whitney. Real-time detection and tracking of vehicle base fronts for measuring traffic counts and speeds on highways. *Transportation Research Board Annual Meeting*, 2007.

[20] C. Mallikarjuna, A. Phanindra, and K. R. Rao. Traffic data collection under mixed traffic conditions using video image processing. *Journal of Transportation Engineering*, 2009.

[21] C. Mallikarjuna and K. R. Rao. Heterogenous traffic flow modeling: a complete methodology. *Transportmetrica*, 2011.

[22] C. Mallikarjuna, K. R. Rao, and N. Seethepalli. Analysis of microscopic data under heterogenous traffic conditions. *Transport*, 2010.

[23] D. Merugu, B. S. Prabhakar, and N. S. Rama. An Incentive Mechanism for Decongesting the Roads: A Pilot Program in Bangalore. In *NetEcon*, 2009.

[24] Personal communication, Mapunity, 2012.

[25] A. Quinn and R. Nakibuule. Traffic flow monitoring in crowded cities. In *AAAI Spring Symposium on AI for Development*, 2010.

[26] R.Balan, N.Khoa, and J.Lingxiao. Real-time trip information service for a large taxi fleet. In *Mobisys*, Jun 2011.

[27] A. Salim, L. Vanajakshi, and S. Subramanian. Estimation of average space headway under heterogeneous traffic conditions. *International Journal of Recent Trends in Engineering and Technology*, 2010.

[28] R. Sen, A. Maurya, B. Raman, R.Mehta, R.Kalyanaraman, S.Roy, and P. Sharma. Kyunqueue: A sensor network system to monitor road traffic queues. In *Sensys*, Nov 2012.

[29] R. Sen, B. Raman, and P. Sharma. Horn-ok-please. In *Mobisys*, June 2010.

[30] R. Sen, P. Siriah, and B. Raman. Roadsoundsense: Acoustic sensing based road congestion monitoring in developing regions. In *IEEE SECON*, June 2011.

[31] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Sensys*, November 2009.

[32] Y. Xiaodong, X. Ping, D. Lingyu, and T. Qi. An algorithm to estimate mean vehicle speed from mpeg skycam video. *Multimedia Tools Appl.*, 2007.

[33] J. Yoon, B. Noble, and M. Liu. Surface street traffic estimation. In *MobiSys*. ACM, 2007.