Horn-Ok-Please

Rijurekha Sen, Bhaskaran Raman and Prashima Sharma
Dept. of Computer Science, IIT Bombay
riju@cse.iitb.ac.in, br@cse.iitb.ac.in, prashima@cse.iitb.ac.in

ABSTRACT

Road congestion is a common problem worldwide. Existing Intelligent Transport Systems (ITS) are mostly inapplicable in developing regions due to high cost and assumptions of orderly traffic. In this work, we develop a low-cost technique to estimate vehicular speed, based on vehicular honks. Honks are a characteristic feature of the chaotic road conditions common in many developing regions like India and South-East Asia.

We envision a system where dynamic road-traffic information is learnt using inexpensive, wireless-enabled on-road sensors. Subsequent analyzed information can then be sent to mobile road users; this would fit well with the burgeoning mobile market in developing regions. The core of our technique comprises a pair of road side acoustic sensors, separated by a distance. If a moving vehicle honks between the two sensors, its speed can be estimated from the Doppler shift of the honk frequency. In this context, we have developed algorithms for honk detection, honk matching across sensors, and speed estimation. Based on the speed estimates, we subsequently detect road congestion.

We have done extensive experiments in semi-controlled settings as well as real road scenarios under different traffic conditions. Using over 18 hours of road-side recordings, we show that our speed estimation technique is effective in real conditions. Further, we use our data to characterize traffic state as free-flowing versus congested using a variety of metrics: the vehicle speed distribution, the number and duration of honks. Our results show clear statistical divergence of congested versus free flowing traffic states, and a threshold-based classification accuracy of 70-100% in most situations.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]; C.2.4 [**Computer-Communication Networks**]

General Terms

Design, Experimentation, Performance

Keywords

ITS, sensor network, audio signal processing

1. INTRODUCTION

The issue of road traffic congestion is an important one in most places of the world today. The problem is especially severe in developing regions like South/South-East Asia, where new-found wealth for a section of the population has driven traffic congestion to the brink in most cities [1]. The road traffic in cities like Bangalore is alarming, with over 5 million vehicles plying on barely 3000 kms of road [2]. Growth of infrastructure has not been adequate due to a variety of reasons, including insufficient funds, bureaucracy, and sheer lack of physical space for the traffic volume.

The issue needs specific attention in developing countries not only because the severity of the problem, but also because the nature of traffic is fundamentally different from that in the developed world. The difference needs to be experienced to be fully understood, but an appreciation can be partially gleaned from the representative videos at [3, 4]. Unlike traffic in developed countries, traffic on city-roads in many developing regions is characterized by two aspects (1) There is high variability in size and speed of vehicles. The same road is shared by 4-wheeled buses and trucks, 4-wheeled cars and vans, 3-wheeled vans and auto-rickshaws, 2-wheeler motor-bikes, bicycles, often-times pedestrians and bullock-carts too. (2) Partly as a corollary of the variability, traffic is often chaotic, with no semblance of a lane-system common in developed countries [5, 3, 4].

Intelligent Transportation Systems (ITS) refers to a host of techniques using sensors to alleviate road traffic congestion. But most sensing techniques like inductive-loops, magnetic detectors, or imaging-based techniques not only have a high cost [6], but also make assumptions of orderliness or lane-systems or low vehicle variability, which are inapplicable in the chaotic road conditions [1] prevalent in most developing regions.

We envision a system where inexpensive, wireless-enabled, on-road sensors are deployed widely, to learn and report dynamic road traffic information. Subsequently analyzed information, in the form of useful traffic updates, is sent to road users over their mobiles. This fits in well with the burgeoning mobile-phone market [7], and the budding mobile-data market [8] in the cities of developing regions.

To this end, this paper presents a novel, inexpensive technique for sensing vehicular speed. Subsequently, we present techniques to classify the traffic state as congested versus free-flowing. To estimate vehicle speeds, we use *vehicular honks*, a prevalent feature in chaotic road conditions (see [3, 4]). For instance, on Indian roads (city-roads as well as highways), honks are common under all road conditions: slow or fast, free-flowing or congested. In fact, honks are deeply inter-twined with the on-road driving protocol, so much so that honks are often required for "safe" driving (i.e. other drivers & pedestrians expect honks). The title of this paper is a phrase painted behind almost every truck/van in India [9]; there is no better proof of how deeply entwined honks are, in Indian road driving protocol. In this work, we put this otherwise negative feature of chaotic road traffic to positive use. While our narrative and experiments are necessarily India-focused, we believe that the technique is applicable in other developing regions with chaotic traffic too, such as South and South East Asia.

Our technique uses a pair of low-cost audio sensors deployed on the road-side, and is based on the Doppler shift of *vehicular honks*, to estimate vehicular speed. Doppler shift based speed estimation itself is of course very well known (radars use this principle); the novelty and usefulness of our work lies in applying this for vehicular honks. The use of acoustic sensors means that the hardware we require for our sensors is the same as any mobile phone; thus our technique also has the advantage of riding the low-price-curve of the mobile market.

The contributions in this paper are as follows. (1) We present the novel idea of using vehicular honks, a prevalent feature on Indian roads, to gauge vehicle speed. (2) We develop an inexpensive two-sensor architecture to implement the above idea. (3) We develop algorithms for practical honk detection, honk matching across sensors, and frequency extraction for speed estimation. We use extensive on-road experiments in this process. (4) We present over 18 hours of data collected on different roads to show the usefulness of the speed estimation algorithm in road congestion detection.

Our results show that the speed estimation technique based on honks is practical under real city-road conditions. And further that the estimated speeds can be used to clearly distinguish between various traffic states. The threshold-based classification shows as high as 75-100% match with ground-truth on real roads. This thus holds enormous promise for widespread practical deployment.

The rest of the paper is organized as follows. Sec. 2 describes related work. We then describe our overall architecture and the envisioned context of usage, in Sec. 3. We develop the details of our honk detection, honk matching, and speed estimation algorithms in Sec. 4. Sec. 5 presents the evaluation of our speed estimation technique. Subsequently, Sec. 6 focuses on using the speed estimates to classify road traffic state as congested versus free-flowing. The paper concludes in Sec. 7.

2. RELATED WORK

We now discuss the state-of-the-art in related literature, under various categories.

2.1 Existing on-road sensing techniques:

Various on-road sensing techniques are deployed in western cities. For instance, pairs of inductive loop detectors can be used to identify vehicles based on their length [10]. This technique is too expensive (several thousands of U.S.$ per installation) for widespread deployment and maintenance even in developed countries [6]. Furthermore, the inherent assumption of lane-based orderly traffic makes it inapplicable for chaotic road conditions. Similar criticisms apply for imaging-based sensing [11] techniques too [6, 1], with costs running into $10-20K per installation. While magnetic sensor-based solutions [12] can be relatively inexpensive, they also make assumptions of traffic orderliness [1]. Furthermore, the technique is unreliable for motorcycles [12], which form a substantial part of road traffic in developing regions.

2.2 Probe-based techniques:

Given the costs of the above on-road sensing techniques, the work in [6] considers GPS-enabled probe-vehicles. Using probe-vehicles' GPS traces, they first classify the road network into *segments* delimited by traffic signals. Temporal and spatial speed traces within each segment are then analyzed, and a thresholding technique is developed to categorize traffic within the segment as congested versus free-flowing. Such probe-based techniques are more applicable to developing regions due to the lower cost, and lack of traffic orderliness assumptions. However, various Indian city roads have a large fraction of signal-less intersections, where drivers follow a random protocol to pass the intersection (see [4]). Even when there are traffic signals, it is not uncommon for a large fraction of vehicles to violate it. These aspects place a significant question mark on the applicability of the techniques developed in [6] for chaotic road conditions.

2.3 Techniques in chaotic road conditions:

Nericell [5] represents one of the early works in developing techniques specifically for chaotic road conditions. It uses sensors in high-end mobile phones, such as microphones, accelerometers and GPS, belonging to users traveling in cars to detect honks, potholes in roads, and vehicle braking. We use the honk detection mechanism in [5] as a starting point and enhance it further. Our technique itself is quite different however: we use on-road sensors to detect vehicle honks, and use Doppler shift to estimate vehicle speed. We require only relatively inexpensive audio sensors (microphone). Also, in comparison to [5], our work includes the additional aspect of classifying traffic state as congested versus free-flowing.

2.4 Other audio-based techniques:

The use of Doppler-based speed estimation is quite well known. Radars are based on this principle, and the adaptation of the technique to police "speed-guns" is common. Radars require the sound beam to be "aimed" at a specific moving vehicle. On a road where there are multiple vehicles of various sizes (i.e. multiple sources of reflection), and where the ambient noise is high, the use of radars is questionable. Indeed, we are unaware of the use of radars on Indian roads. Unlike radars, in this paper, we use honk sounds originating from moving vehicles themselves.

The work in [13] uses signal processing techniques to estimate vehicle speed based on the Doppler shift of engine and wheel noise. Since the technique assumes that a particular recording belongs to a single vehicle, its applicability in a setting, where we have a mix of sounds from various vehicles of different sizes and speeds, is questionable.

3. OVERALL ARCHITECTURE

The system we envision comprises of inexpensive road-side sensors, collecting dynamic information about vehicle movement on the roads. The sensors are wireless enabled, and communicate with a central server to convey the learnt information. This is shown in Fig. 1. Subsequent analysis is used to extract information such as the road traffic state, and this is conveyed to other mobile users. The traffic state can be in terms of a simple free-flowing versus congested classification, or finer grained.

In this context, this paper focuses on a low-cost approach to use *vehicular honks*, and their Doppler shift to estimate vehicle velocity. We propose to use audio sensors (microphones) in this process.

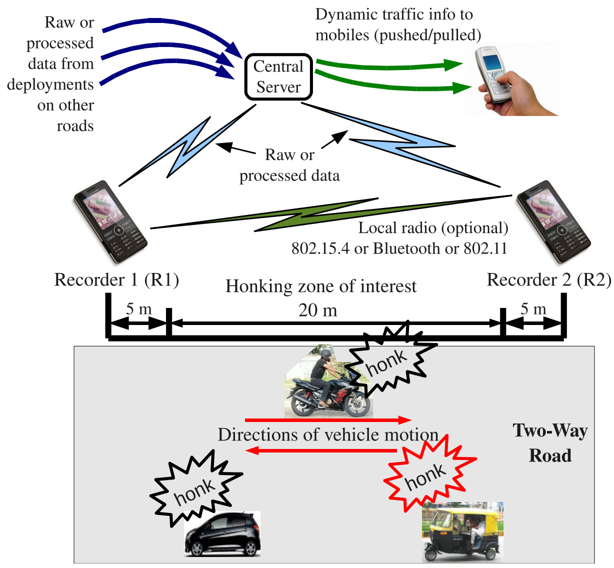

Figure 1: System Architecture

3.1 Using Doppler shift:

Suppose that a sound source moves with speed v_s, and the receiver (observer) is stationary. Denote the emitted audio frequency as f_0 and speed of sound as v. When the source moving away from the receiver, the frequency observed at the receiver is given by,

$$f_1 = \frac{v}{(v + v_s)} f_0 \tag{1}$$

And when the source moving towards the receiver, the frequency observed at the receiver is given by,

$$f_2 = \frac{v}{(v - v_s)} f_0 \tag{2}$$

3.2 Two-sensor architecture:

If f_0 is known, v_s can be estimated easily from Eqn. 1 or Eqn. 2, and one sensor would suffice. But it is not easy to guess f_0, as different honks have different base frequencies. We thus use a two-sensor architecture: Fig. 1 depicts a deployment of two wireless-enabled audio sensors (recorders) by the side of a two-way road. When a moving vehicle blows honk in between the two receivers, it is approaching one receiver and receding from the other. Substituting the value of f_0 from Eqn. 1 in Eqn. 2, we get following equation,

$$v_s = \frac{-(f_1 - f_2)}{(f_1 + f_2)} v \tag{3}$$

The above approach can not only compute the speed but also the direction of motion, on a two-way street. The steps involved in speed estimation are as follows.

1. **Honk detection:** The two sensors (recorders) record and *detect* the honk sample independently.
2. **Honk matching:** We then have to *match* honks with each other, so that we apply Eqn. 3 for the same honk.
3. **Frequency extraction:** We have to *extract* f_1 and f_2 and apply Eqn. 3 to get the speed estimate.

The second and third steps can be done at one of the recorders, or at a central server. If done at one of the recorders, the final speed sample can be communicated to the central server. This is shown in Fig. 1.

The honk-matching step requires that the two recorders be time-synchronized. This can be done using the wireless connection to the central server, or using a local radio such as Bluetooth, 802.15.4, or 802.11. For communication with the central server, we could use GPRS/3G or SMS. We could even use dpipe [14], via a local radio. Fig. 1 shows a particular two-sensor deployment feeding data to the central server. The central server also receives similar measurements from other similar deployments at other locations in the road network of interest.

3.3 Line of vehicle motion:

In our architecture, we assume that the line of vehicle motion coincides with the line joining the two sensors. This causes some inaccuracy, but there are several ways to reduce it. (1) Most city roads are at most two "lanes" wide, or about 5m each way. This reduces the inaccuracy as the inter-sensor distance is large with respect to the road width (2) Our algorithms seek to restrict honk samples to a sub-region near the middle of the two recorders; we call this the "honking zone of interest" (see Fig. 1). The intuition behind this is that near the middle of the two recorders, the speed estimate inaccuracy due to distance between the line of motion and the line joining the recorders, is minimized. (3) Many roads have a divider separating the two directions of traffic. In such cases, the pair of recorders could be deployed on the divider, and not on the side of the road.

Despite the above measures, some inaccuracy is unavoidable. But as we show, this inaccuracy does not matter when we finally estimate the traffic state.

3.4 Sensor placement:

There are several issues related to where the sensors are to be placed. The two sensors need to be sufficiently far away from one another for the primary reason that we get sufficient honk samples in-between. An additional reason may be the reduction of the above-mentioned inaccuracy. However, if the two sensors are too far apart, the chances that the same honk is heard at both places reduce. Furthermore, if a local radio is used for communication between the two sensors, its range is also a concern.

We have chosen an inter-sensor distance of 30m, and a 20m long honking zone of interest (see Fig. 1). This setting gives a good number of honks in the zone of interest. And if the sensors are mounted on light poles, the local radio range can be several tens of meters if not more, even for the relatively high frequency of 2.4GHz [15] for 802.15.4 or Bluetooth or 802.11.

The basis of our architecture dictates that the pair of sensors must be deployed where there is a clearly defined line of traffic motion (in either direction). In other words, there must be no nearby side-roads or cross-roads or intersections, since honk samples from such settings would not have a well-defined line of motion.

3.5 Advantages of our approach:

There are a whole host of advantages to our approach of using honks. (1) First and foremost, honks are a natural part of chaotic traffic, since honks are used as a warning to avert collisions or indicate impatience. As already noted, honks are common on Indian roads, under all conditions, on most roads. So our method is an excellent fit for chaotic roads. (2) The number of speed samples is likely to be far higher than any probe-vehicle based mechanism. Furthermore, we readily get speed samples from all kinds of vehicles on the road (4-wheelers, 3-wheelers, 2-wheelers, etc.). (3) The more used or congested a road is, the more the reason to honk; indeed we observe this consistently in our experimental data. So we have a nice property: more the need for traffic updates, more are the vehicular speed samples we get. (4) Honks are used to warn other drivers; so by their very design, honks are easily distinguishable from other road noise. (5) For a similar reason, unlike other road noise, most honks are non-overlapping in time, across vehicles (except in very high congestion); so no sophisticated sound separation algorithm is necessary. (6) Last but not the least, the wireless-enabled sensors are cheap. The hardware we require is exactly what is present in a commercial mobile phone (this is why Fig. 1 depicts the recorders as phones). Hence we can ride the price-curve of mobile phones: in developing regions like India, one can get mobile phones for as cheap as $20.

3.6 Scope of this paper:

While Fig. 1 gives the overall context, this paper itself focuses on the main unsolved challenges related to (a) the 3-step speed estimation and (b) its subsequent use in traffic state classification. Specific aspects we *do not* address in this paper are: (a) determining the set of locations in a road network, at which to install pairs of recorders, (b) using the collection of traffic state reports from different installations to estimate metrics such as travel time. These are interesting avenues for future work.

3.7 Challenges:

In our honk-based approach, there are several open questions. Are there sufficient honks in practice? Can they be detected and matched across sensors in the presence of road noise, multiple random sound reflections (echoes), and other sources of inaccuracies? What might be the honk detection, matching, and frequency extraction algorithms? How accurately can vehicle speeds be estimated? Given various vehicle speed estimates, can we indeed distinguish between congested and free-flowing traffic states? Can we distinguish traffic conditions between two directions on bidirectional roads? Can we detect the time when congestion starts setting in? We now turn to address these questions methodically.

4. ALGORITHM DESIGN

This section focuses on the three-steps in vehicle speed estimation: honk detection (Sec. 4.3), honk matching (Sec. 4.4), and frequency extraction (Sec. 4.5). Before presenting our algorithms, we first describe our experimental methodology (Sec. 4.1), and present some preliminary honk properties (Sec. 4.2).

4.1 Experimental Methodology

We have taken an experiment driven approach to algorithm development. This is because many algorithm choices become clear only after practical testing. During the experiments, for the recordings, we have used the voice recorder software on Nokia N79 mobile-phones. We have used 16KHz audio sampling. As we shall see shortly, the frequency range of honks is within 2-4KHz, so 8KHz sampling would be enough according to Nyquist's criterion. The criterion states that if a function x(t) contains no frequencies higher than B hertz, it is completely determined by giving its ordinates at a series of points spaced 1/(2B) seconds apart. We double the sampling frequency to reduce noise. We use mono-channel, 16-bit recording: stereo channel or higher bit encoding do not add any benefit to our analysis.

When two (or more) recorders are involved in an experiment, the recorders need to be time-synchronized. For simplicity, we have used a known sound pattern for such synchronization. We record this pattern in each phone at the beginning of recording and clip the file in the recorder which started earlier. The estimated error in synchronization is under a milli-second, which suffices for our algorithms.

We have used two kinds of experimental settings, which we call *campus-road* and *city-road*. The *campus-road* experiments are within the IIT-Bombay campus, where there is relatively little traffic. So we use a motorbike and control when we honk. We however have no control over the frequency pattern, the sound echoes, etc. So the campus-road experiments are semi-controlled. This greatly helped us during the algorithm development process.

The *city-road* experiments are on various city roads. We term one set of roads as *Hira*, which are from a residential locality called *Hiranandani*. These roads were one-lane in each direction, and about 5-6m wide overall. We also had a set of measurements on a much wider road, called *Adi Shankaracharya Marg*, which we abbreviate as *Adi*. This road is 3 "lanes" each way: see [3]. Both *Hira* and *Adi* are known for their congestion at peak times, the latter more so than the former.

4.2 Empirical Data on Honks

In this section, we seek answers to three important questions: (1) are there indeed enough honks? (2) what are typical honk durations? (3) and finally, what are the audio frequencies of interest? Answers to these subsequently guide our algorithm design.

We performed several road-side recordings at *Hira*, using an N79 mobile-phone. The recordings are in terms of 10-minute clips. We recorded in various conditions (morning, noon, evening, night), and at different roads in *Hira*. Since this was a precursor to our honk detection algorithm, we sought to detect honks "manually", using a two-step process, to establish ground-truth. We first look for dark regions in the spectrogram of the recording; such dark regions indicate high amplitude. We use *Praat*, an audio signal processing software, for this. An example is shown in Fig. 2. We then verify that this is indeed a honk by hearing the identified region of recording. The dark region also gives us a measure of the honk duration, within an estimated error of a few milliseconds. We can only guess the error here, since we are *determining* ground-truth.

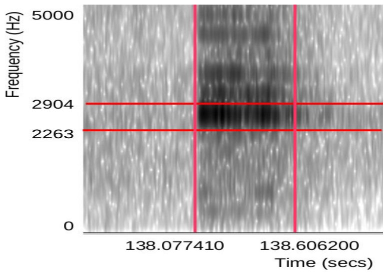

Figure 2: Spectrogram in *Praat*

4.2.1 How often do vehicles honk?

For the 18 ten-minute clips we recorded, we found an average of 30 honks per clip. The median was 27 honks, the minimum 15, and the maximum 63 honks per clip. Note that these honks were those within the recording range of the recorder we used. While these numbers can clearly vary with the road and the conditions, there appear to be a large enough number of honks to get several vehicle speed samples per minute.

4.2.2 How long do vehicles honk?

Fig. 3 shows the CDF of the honk durations, as visually detected in the spectrogram, for the 18 x 10-min = 3 hours of recordings. We see that over 90% of the honks are at least 100ms long. The median honk length is about 200ms. And there are some honks which are more than 1-2 seconds long.

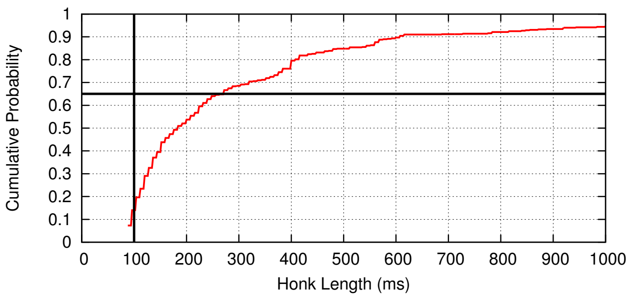

Figure 3: CDF of honk length

4.2.3 What is the audio frequency of honks?

We use the Discrete Fast Fourier Transform (FFT) [16] tool in the *Audacity* software to determine the honks' dominant audio frequencies. Nericell [5] claims that honk frequency range is between 2-4 KHz. We verify this claim in our data: out of approximately 300 honks in the recordings, only 3 have a dominant frequency outside of this range.

4.3 Honk Detection

The first of our three-step speed estimation process is *honk detection*.

4.3.1 Goal

Here, we not only have to detect honks in presence of noise, but also determine each honk's boundary (start and end) in time.

4.3.2 Approach

Nericell [5] uses the following simple honk detection algorithm. The recording is broken up into 100ms windows. A Discrete Fast Fourier Transform (FFT) [16] is performed on each window. A discrete FFT transforms a sample set in time domain to frequency domain. A 100ms window is said to be a honk if there are at least two *spikes*, with at least one spike in the 2-4KHz range. A spike is defined as a frequency whose amplitude in the FFT is at least a threshold T times the average amplitude across all frequencies. Values of 5-10 are reported to work well for T.

While we use Nericell's basic approach, we adapt it in several subtle yet significant ways.

4.3.3 Band-pass filtering:

First, we found in our road experiments that band-pass filtering is a necessary step, to remove noise, especially in the road *Adi*. So we band-pass and filter out (i.e. reduce the amplitude of) sound outside the 2-4KHz range. Such band-passing allows us to have uniform comparison thresholds in all situations, irrespective of road noise.

4.3.4 Breaking time into small windows:

A more important aspect is that the algorithm in [5] only tells whether or not there is a honk in a 100ms window. We need to know the *start* and *end* boundaries of the honk, as accurately as possible. Since propagation delay of sound causes the two recorders to see time-delayed versions of the same honk, honk boundary detection is important for honk matching. And for frequency extraction, accuracy in honk boundaries is necessary for accurate estimates of f_1 and f_2.

For precise honk boundary detection, we need to use small time windows. Now, the number of samples in a time window, to be used in FFT computation of that time window, has to be equal to the number of FFT points. That is, the number of points in frequency domain is the same as the number of samples in time domain. For a given sampling frequency (16 Khz in our case), the number of samples in a time window is directly proportional to the size of the time window, which we want to decrease. Thus a small time window means a reduction in the number of FFT points, and thus the frequency resolution. That is, we cannot have accuracy in the time and frequency domains simultaneously.

This is precisely the reason why we have a separate frequency extraction step. In the honk detection step, where frequency resolution is not important, we choose higher time resolution. In the later frequency extraction step, we focus on better frequency resolution.

For honk detection, we choose to use 128 FFT points (128 is the minimum FFT points supported by the open source FFT implementation we use). With a 16Khz sampling frequency and 128 samples per time window, we have time window size as 8ms which gives good time accuracy.

4.3.5 Algorithm choices:

We considered three possible choices for the algorithm.

1. **PeakVsAvgAllFreq:** This algorithm, similar to [5], considers a time window to be a honk if a frequency in

the 2-4 KHz range has an amplitude at least T times the average of *all frequencies* in that time window. We use a time window of 8ms, and have found that $T = 10$ works uniformly well for all roads, *after* the band-pass filtering step; without the band-passing, we were unable to find one uniform threshold for all situations.

2. **PeakVsAvgHonkFreq:** This algorithm is similar to *PeakVsAvgAllFreq*, except that we compare the peak against the average of the amplitudes in the honk frequency (2-4KHz).

3. **PeakAbsAmp:** This labels an 8ms window as a honk if the absolute threshold of any frequency in 2-4KHz range exceeds -20dB.

4.3.6 Experimental evaluation of algorithm choices:

To evaluate the above algorithm choices, we use the same 3 hours of road-side recording as given in Sec. 4.2, where we manually (visually and through hearing) labeled about 300 honks. A false-positive is an 8ms window which is labeled as not a honk in the ground-truth, but is detected as a honk by the algorithm. And a false-negative is a window which is labeled as a honk in the ground truth, but not detected by the algorithm.

Table 1 tabulates the results for the three algorithms. As we can see from the first row, the initial results are quite poor.

Honk length bounding: On closer look, we found that most of the false positives were due to stray windows. Since our CDF in Fig. 3 shows that over 90% of the honks are longer than 100ms, we use this as a lower-bound in our honk boundary detection. That is, any 8ms window which is not part of a train of at least 14 such windows, is classified as *not* a honk. This lower-bounds the honk length to be at least $14 \times 8ms = 112ms$. The second row in Tab. 1 shows the effect of honk length bounding.

Honk merging: Furthermore, in our various in-campus experiments, we found that the honk detection algorithms many times *split* the same honk as several shorter honks. To correct this, we introduced a *merging* step, where two trains of 8ms windows (detected as honks) are merged if they are separated by not more than 3 intervening non-honk 8ms windows. The last row in Tab. 1 shows the effect of this merging step. We see that the false negatives come down further, with almost no effect on the false positive rate.

More than the reduction in the false negative rate, honk merging ensures that we do not have spurious honk boundaries (start/end), which is important for honk matching, as we shall see.

4.3.7 Algorithm choice:

PeakVsAvgHonkFreq has a high rate of false negatives. The reason is, in a honk window, most frequencies in honk range have fairly high amplitudes. So the peak cannot exceed the average amplitude of the honk frequency range by a threshold. The other two algorithms have comparable performances, with *PeakVsAvgAllFreq* being the better of the two. So we choose *PeakVsAvgAllFreq* as our honk detection algorithm.

	PeakVsAvgAllFreq		PeakVsAvgHonkFreq		PeakAbsAmp	
Stage	fp (%)	fn (%)	fp (%)	fn (%)	fp (%)	fn (%)
Default	22.3	0.2	2.3	61	18.9	0.3
length bounding	5.6	0.7	0.03	93.8	10	1.04
honk merging	5.7	0.4	0.03	93.7	10.3	1.01

Table 1: Comparison of honk detection algorithms

The final honk detection algorithm: *(1) Perform band-passing to filter out (reduce the amplitude of) sounds outside 2-4KHz. (2) Break time into 8ms windows, and use* **PeakVsAvgAllFreq** *(with $T = 10$) to classify each window as a honk or non-honk. (3) Use honk length lower bounding followed by honk window train merging to arrive at the final set of honks, along with their time boundaries.*

4.4 Honk Matching

The second of our three-step speed estimation process is *honk matching*.

4.4.1 Goal

Honk detection can be done independently by each recorder. After detection, the same honk has to be *matched* across the two recordings. In our honk-matching step, we also seek to ensure that we match only honks in the "zone of interest" (Fig. 1).

4.4.2 Intuitions

To match honks, we consider the following two intuitions.

- **StartTimeDiff:** For two honk windows h_1 and h_2, at recorders R1 and R2 respectively, to have originated from the same honk, within the zone of interest, the difference between the start times of h_1 and h_2 must be bounded. For instance, in Fig. 1, suppose the honking vehicle is at distance x_1 and x_2 respectively from the two recorders, when it starts honking. And if the vehicle is within the zone of interest at this time, then $|x_1 - x_2| < 20m$. So ideally, the start times of h_1 and h_2 must differ by not more than $D = \frac{20}{v}$, where v is the speed of sound.

- **DurnRatio:** This criterion bounds the ratio of honk durations in the two recorders to be below R. Ideally, if d_1 and d_2 are the honk durations at recorders R1 and R2 respectively, $d_1 f_1 = d_2 f_2$, since the number of wavelengths (lambdas) seen by both the recorders is the same (also same as the number of wavelengths generated at source). Here we are ignoring the change in vehicle position for the duration of the honk. So, $\frac{d_1}{d_2} = \frac{f_2}{f_1} = \frac{v+v_s}{v-v_s}$ where v is speed of sound and v_s is speed of vehicle. Since v is fixed, this ratio will increase with increasing v_s. If we take maximum value of v_s to be 54Km/h i.e. 15m/sec, which is realistic for most city roads, ideally $R = \frac{v+15}{v-15}$.

4.4.3 Sources of error:

There are two main possible sources of error. First, there may be environment-dependent echoes. The second source of error is something we realized after experimenting: the honk amplitude is different at the two recorders. This is especially so when the vehicle is in-between the two recorders:

most honk installations are directional by design. That is, they give a higher amplitude in front of the vehicle than behind it. Such amplitude difference in turn means that one recorder will *detect* it earlier than the other, for any given value of T in our detection algorithm.

4.4.4 Experimental evaluation of honk matching heuristics:

We use semi-controlled *campus-road* experiments to test the usability of *StartTimeDiff* and *DurnRatio*. For this, we place Recorder-1 near a stationary bike. This is shown in Fig. 4. Recorder-2 is first at a distance of 10m and then at a distance of 20-m from Recorder-1. For the first position of Recorder-2, we blow the bike honk 15 times, for the second position 10 times and record in both the recorders. This setup allows to examine how *StartTimeDiff* and *DurnRatio* fare.

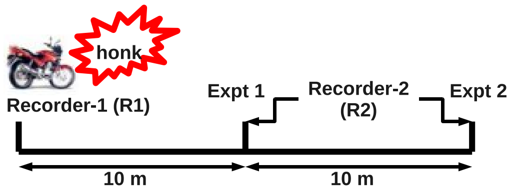

Figure 4: Evaluation setup for StartTimeDiff & DurnRatio

Verifying StartTimeDiff: For sound speed of $v = 340m/s$, the expected start time difference is $29ms$ at 10m and $59ms$ at 20m. We measure the actual start time difference for the 25 honks recorded in the above experiment using our honk detection algorithm. Fig. 5 shows the results.

We can see that most of the start time differences are close to what we expect. But there can be errors as much as a few tens of milli-seconds, due to the various reasons listed earlier. Given this experiment, we take the *StartTimeDiff* threshold value of $D = 80ms$, keeping some allowance from the expected value of $59ms$ at 20m.

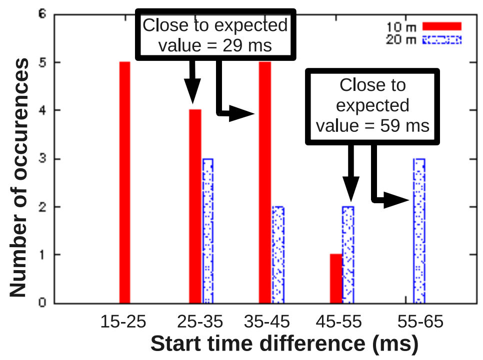

Figure 5: Start time difference values (ms) for 25 honks

Verifying DurnRatio: To evaluate the DurnRatio heuristic, we calculate the durations of the 25 honks using our detection algorithm. The speed of the bike being 0, the durations of the same honk in the two recordings, should be the same; i.e. we expect $d_1 = d_2$, or $\frac{d_1}{d_2} = 1$. But at a distance of 10m, we found that $\frac{d_1}{d_2}$ varied all the way from 0.43 to 1.75 for the 15 honks. At a distance of 20m, the values varied from 0.38 to 0.96. In both cases, most values were significantly different from the expected value of 1.

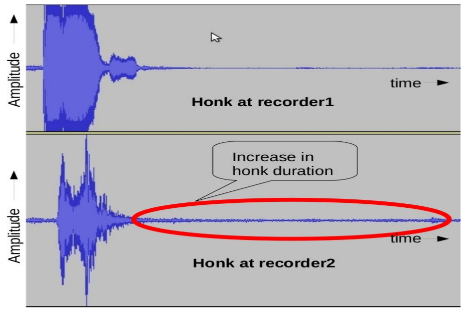

Figure 6: Trailing honk pattern in Recorder-2

We viewed each honk pair in *Audacity* and found a significant trailing pattern after each honk in Recorder-2 (see Fig. 6). This is likely due to echoes. The cases where $d_1 > d_2$ are likely due to the fact that the honk source was near Recorder-1. Since there is no discernible pattern to the variation of $\frac{d_1}{d_2}$, we decide not to use it at all in the honk matching algorithm.

The final honk matching algorithm: is thus as follows. *If the start time of a honk (h1) recorded in one recorder is greater or less than the start time of a honk (h2) recorded in second recorder by at most $D = 80ms$, h1 and h2 are matched; i.e. taken to be from the same honk.*

4.5 Frequency Extraction

The final of our three-step speed estimation process is *frequency extraction.*

4.5.1 Goal

Here we extract frequencies f_1 and f_2 from a pair of matched honks to calculate speed using them.

4.5.2 Choosing FFT points:

For honk detection, we used 128-point FFTs, since we needed good time resolution. Here we need good frequency resolution. The frequency resolution for N-point FFT is given as $n = F/N$, where $F = 16KHz$ is the sampling frequency. In other words, the error in frequency estimation can be as high as $n/2$ in the worst case. A higher N thus means a lower n and hence a lower error in the final speed estimate.

In choosing high value of N, we have two criteria mandated by the FFT computation (a) N should be a power of 2, and (b) each time window passed to the FFT computation algorithm should have N samples. If we choose $N = 4096$, we need time window of 256ms as our sampling frequency is 16KHz. From Fig. 3, 70% of the honks in each sound clip is less than 250ms in length, so we will be discarding most honks if we stipulate a 256ms time window. Hence we choose $N = 2048$, which needs a 128 ms time window. If a honk has many 128ms windows, then we do independent 2048-point FFTs in each window, and average out the amplitude for each frequency across the multiple windows.

According to Sec. 4.3, the minimum honk duration for us is 112ms. So for the few honks with duration 112 ms or 120 ms (our honk duration always is a multiple of 8 as detection uses time window of 8 ms), we use $N = 1024$.

4.5.3 *Choosing frequency peaks:*

From a pair of honks, matched across the two recorders, we do an N-point FFT, N chosen as above. Using various *campus-road* experiments' data, and observing the FFT of the matched honks in *Audacity*, we find the following. In most cases, there is a close correspondence between the local maximas (in terms of amplitude) of frequencies in either recording. This is shown in Fig. 7. That is, local maximas in the original sound show up as local maximas even after Doppler shift. This is intuitive, since the Doppler phenomenon is not concerned with the amplitude, and since attenuation is similar across frequencies in the region of interest.

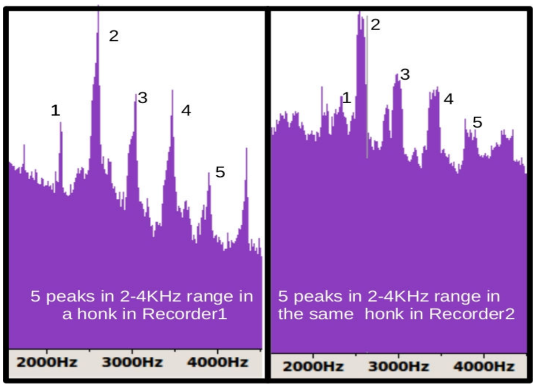

Figure 7: Close correspondence between local maximas in the two recordings

We thus use the following heuristic, termed **SinglePeak**. From each honk, we choose the frequency with the highest amplitude in the FFT, and use these as f_1 and f_2 for speed estimation.

We used several *campus-road* experiments to evaluate the effectiveness of the above mechanism. From 80 pairs of matched honks from these experiments, we found that for 75 pairs, the speed estimates were fairly accurate. But in the remaining 5 cases, we saw huge errors, such as $100Km/h$. On closer examination of the recordings, we found that the highest peak in one recording corresponded not to the highest, but to the second highest peak in the other recording. This is shown in Fig. 8.

We thus correct *SinglePeak*, and use the following heuristic termed **TwoPeak**. This corrected heuristic uses the following observation. Without loss of generality assume that $f_1 < f_2$. So $\frac{f_1}{f_2} = \frac{v-v_s}{v+v_s}$, which lower for higher v_s. With $v = 340m/s$, and v_s being the vehicle speed, we can lower bound $\frac{f_1}{f_2}$ by upper-bounding v_s. Assuming an upper bound of 50Km/h, which is practical for most city roads, the lower bound for $\frac{f_1}{f_2}$ is 0.92.

So in *TwoPeak*, we first seek to use the highest amplitude peaks in the two recordings. If this gives a value of $\frac{f_1}{f_2} < 0.92$, then we assume that the local maximas have been exchanged in the two Doppler shifted recordings. We then consider all the other three possible combinations of the highest and second highest peaks among the two recordings.

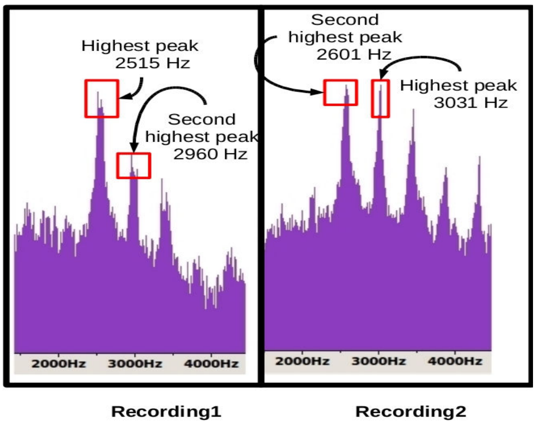

Figure 8: 1^{st} & 2^{nd} peaks exchanged in the two recordings

We take the combination which gives $0.92 \leq \frac{lowerFreq}{higherFreq} \leq 1$ as the final frequencies for speed estimation.

The final frequency extraction algorithm: is thus as follows. *Compute 2048 point FFT for a matched pair of honks, for honk length* $\geq 128ms$. *Compute 1024 point FFT if honk length is between* $112ms$ *and* $128ms$. *Consider frequencies* f_1 *and* f_2 *as per the* **TwoPeak** *heuristic, and use Eqn. 3 for speed estimation.*

5. EXPERIMENTAL EVALUATION OF SPEED ESTIMATION TECHNIQUE

How well does our 3-step speed estimation technique work in practice? We experimentally evaluate this now. We present both *campus-road* and *city-road* experiments. For the *city-road* experiments, we considered both *Hira* and *Adi*. For all the experiments presented in this section, we used our own motorbike and honks from it.

5.1 Initial experiments, the issue of ground truth:

We conducted several initial *campus-road* experiments, where we noted the ground-truth from the vehicle's speedometer. The speed estimated by our algorithm was always within about 5-10Km/h of what we expected. But we quickly realized that the ground-truth in these experiments was always suspect. Knowing the actual speed of the vehicle is difficult, even for the person driving the vehicle. There can be speedometer errors, parallax errors while reading, etc. In many situations, it was even dangerous to divert the driver's attention to the speedometer, even on campus roads. So we did not even attempt this in our *city-road* experiments. Since the ground-truth is inexact, we do not report results from these initial experiments.

Use of a mobile recorder for ground-truth estimation: In our setup, we used the following mechanism to estimate the ground truth. Apart from the on-road recorders, we place a third recorder, called Recorder-3 (R3), on the moving vehicle. Since this recording has no Doppler shift, it should give f_0 as in Eqns. 1 & 2. This procedure for ground-truth has errors too, for instance in estimation of f_0 itself. So for each experiment we have also done a sanity check in terms what speed we expect approximately.

5.2 Campus-road experiments:

On a campus road, our bike was driven past the sensors at various speeds. We varied the speed from 0 Km/h (stationary), to slow (about 10 Km/h), to moderate (about 25 Km/h) to fast (about 35 Km/h). The vehicle blows a honk near the middle of the two recorders. A total of 30 honks are blown in 30 different experimental runs.

5.3 City-road experiments:

We conducted similar experiments at roads *Hira* and *Adi* too. Here too, we varied the motorbike speed between 0 Km/h and about 40 Km/h (the actual speed here was also determined by the traffic situation at that instant). We have 18 honk samples each from *Hira* & *Adi*, making a total of 36 honks. In these experiments, there are several other vehicles' honks too in the same recording. To distinguish our own motorbike's honk from these (which is necessary to evaluate the speed estimation), we annotated the recording by speaking into one of the recorders.

In the campus-road experiments, out of the 30 honks blown, 25 are matched across all the three pairs of recorders, while the remaining 5 are not detected in one of the three recorders. And in the city-road experiments, 4 out of the 36 honk samples were lost due to manual annotation errors. And 26 out of the remaining 32 honks were matched across all the three recorders.

5.4 Results:

We have three estimates of speed: one from Recorder-1 and Recorder-2, using Eqn. 3, which we term v_{12}. We also get an estimate from Recorder-1 and Recorder-3, using Eqn. 2, which we term v_{13}. We get v_{23} similarly from Recorder-2 and Recorder-3, using Eqn. 1. From these, we define three measures of error.

- **(1) Relative Error** - % ratio of v_{12} as estimated speed in numerator and $\frac{v_{13}+v_{23}}{2}$ as actual speed in denominator.
- **(2) Max3Err** - maximum of the three error quantities $|v_{12} - v_{13}|$, $|v_{13} - v_{23}|$, and $|v_{23} - v_{12}|$.
- **(3) Avg3Err** - average of the three error quantities $|v_{12} - v_{13}|$, $|v_{13} - v_{23}|$, and $|v_{23} - v_{12}|$.

For each of the 25 matched honks in *campus-road* experiments, and the 26 matched honks on *city-road* experiments, Fig. 9 shows the measures of error. **Max3Err** is given on the left y-axis and the **Relative Error** is given on the right y-axis. The points on the x-axis are sorted in increasing order of relative error.

Avg3Err is not shown in this graph for clarity of presentation as **Avg3Err** = 2/3 **Max3Err**. The explanation for this is as follows. Without loss of generality, let us assume $0 <= v_{12} <= v_{13} <= v_{23}$. **Max3Err** $= \max(v_{13} - v_{12}, v_{23} - v_{12}, v_{23} - v_{13}) = (v_{23} - v_{12})$. **Avg3Err** $= \text{average}(v_{13} - v_{12}, v_{23} - v_{12}, v_{23} - v_{13}) = 2/3(v_{23} - v_{12})$. Hence the two measures will follow similar pattern.

We see that both in terms of absolute and relative error, our mechanism is quite reliable, even in noisy city road conditions. The *Max3Err* measures are mostly under 5-10Km/h. The relative error is mostly under 10%. There is one case of high relative error of about 65%. We verified that this was a case where the absolute speed itself was low, and hence the relative error is high.

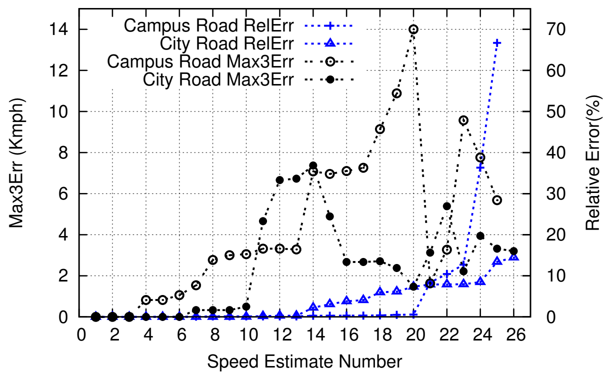

Figure 9: Speed estimate errors

5.5 Varying the position of vehicle honk:

While the above experiments varied the vehicle speed, they kept the honk position fixed (near the middle of the two recorders). We now vary the honk position, on our city-road experiments at *Hira* and *Adi*. We consider 7 different honk positions: this is depicted in Fig. 10. The vehicle moves from position 1 to 7 at a fixed speed (as far as the traffic would allow), and honks approximately at the given positions. Three honk positions, (3,4,5), are between the recorder positions. These 3 are in the honking zone of interest. Two positions, (2,6), are at the two recorders and the remaining two, (1,7), about 10m before and after Recorder-2 and Recorder-1 respectively.

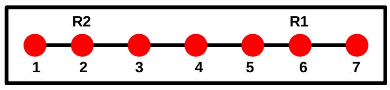

Figure 10: Honking positions of bike

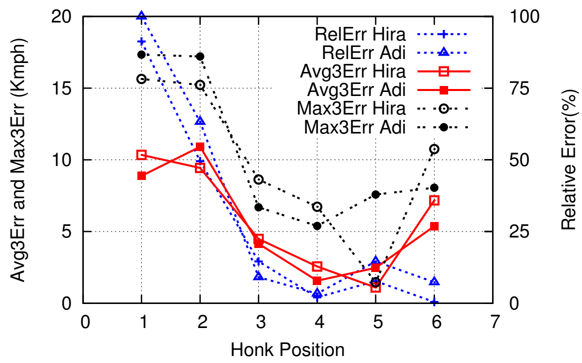

Figure 11: Speed estimate errors at various honk positions

There were a total of 6 honks each at each position except 4, which had a total of 12 honks. At a given position, some honks are matched, while some are not. For each position, Fig. 11 gives the average of the *Avg3Err*, *Max3Err* and relative error measures. The plotted value is averaged across the various number of matched honks for each position. There are no matches at position 7, and hence no data point is shown at that position.

As earlier, the relative error is very low (under 5%) at position 4; it is about 15% for positions 3, 5 and 6. The *Avg3Err* is below 5 Km/h and the *Max3Err* is below 10 Km/h at 3, 4 and 5.

Ideally, our honk matching algorithm should not have matched honks at positions 1, 2, 6, and 7, since the zone of interest is between positions 3 & 5. While position 7 gives no matches, as expected, position 1, 2, and 6 had matched honks. They had 2, 4, and 2 honks matched each, out of a total of 6 honks at each position.

The speed estimates at positions 1, 2, and 6 do show high error. The relative error in speed estimates for positions 1 & 2 are as high as 60-100%. A closer look at the data revealed that these are cases of incorrect low-speed estimates (in fact, zero-speed estimates at position-1), when the honk is outside the zone of interest. These are caused due to false positives in the honk matching step.

5.6 Summary:

To summarize, our speed estimation technique performs well in most situations (in honk positions 3, 4, 5, 6, & 7), both in terms of absolute error as well as relative error. There are however some honk positions (1 & 2) where the speed estimates can be poor due to bad honk matches. In the next section, we shall see how we can work around these, and estimate traffic state despite some fraction of errors in vehicular speed samples.

6. APPLICATION IN TRAFFIC STATE CLASSIFICATION

Given various vehicular speed estimates, can we tell the current traffic state? This would indeed be very useful to on-road commuters, or those planning to commute shortly. In this section, we focus on classifying traffic state into two categories: congested versus free-flowing. While there appears to be promise for a finer grained classification, we leave this for future work.

6.1 Experimental Setup

We performed 18 hours of experiments on city-roads over the month of Nov-2009. Of these 9 hours were in *Hira* and 9 were in *Adi*. We did the experiments in 1-hour chunks, over different days. The times were chosen such that we, by visual observation, were able to clearly classify the ground truth as congested versus free-flowing. Of the 9 hours in *Hira*, 5 were free-flowing and 4 were congested. Even during the 4 hours of congestion, only one side of the road was congested; the other direction was free-flowing. We thus have 9 hours of free-flowing data and 4 hours of congested data from *Hira*.

At *Adi*, we collected 4.5 hours of free-flowing data and 4.5 hours in congested state. The road here was wider, and the road noise so high, that we mostly sense traffic in only one direction, near the side where we placed the sensors. There are almost no honks recorded and matched for traffic in the other direction.

As mentioned earlier, both roads experience heavy congestion during peak times, with the congestion in *Adi* far more severe. *Adi* also has a wider variety of vehicles, large buses and heavy trucks, in addition to two-wheelers, auto-rickshaws and cars, which are prevalent in *Hira*.

6.2 Speed Distribution Plots

Prior to presenting possible metrics for traffic classification, we first get a feel for our data. The primary measurement from a 2-sensor deployment is the set of vehicular speeds. This is what we look at first, from our experiments.

From our recordings, we clip each 1 hour recording into 6 blocks of 10 minutes each. The intuition behind using 10-min chunks is that the underlying traffic characteristic could change significantly from one 10-min period to the next. For each 10-min data, we do honk detection, honk matching and speed estimation from the matched honks, using our algorithms (Sec. 4).

We plot the CDF of speed estimates for each 10-min block. The number of such CDF plots is too many to present here, so we show some representative samples. For instance, Fig. 12 and Fig. 13 show 6 sample CDF plots each ($10min \times 6 = 1hr$ each), under congestion and free-flowing traffic, on *Adi*.

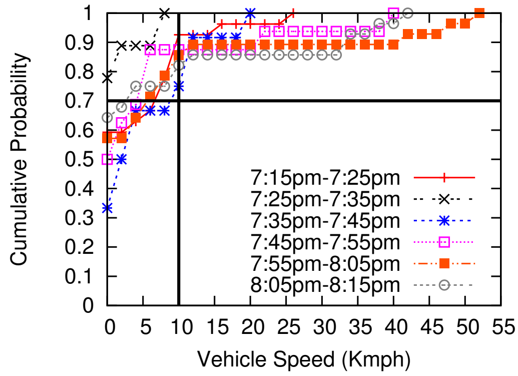

Figure 12: Speed CDF samples: congested traffic in *Adi*

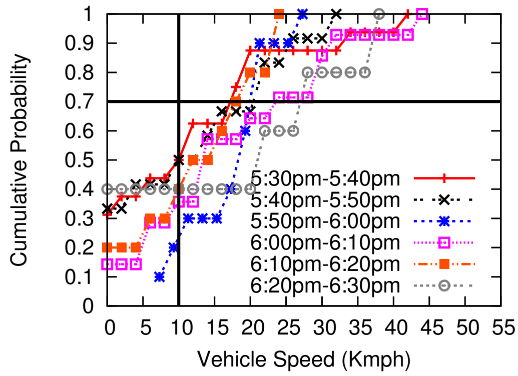

Figure 13: Speed CDF samples: free-flowing traffic in *Adi*

6.2.1 *Observations:*

From the various CDFs of 10-min durations (only 12 of which are shown in Fig. 12 & 13), we observe the following.

1. First, it is striking to see the clear, visually observable difference in the CDFs for the congested versus free-flowing scenarios; we observed this in all of our data.

2. The CDFs under congestion are generally smoother than CDFs under free-flow. This is due to the larger number of speed estimates obtained under congestion.

That is, people honk more under congestion, increasing the number of matched honks.

3. There are a few *high* values of speed under congestion. We manually analyzed the recordings, and identified three possible reasons for this. (a) Many 2-wheelers overtake the stagnant vehicle queue at relatively high speed on the wrong side, sometimes even coming onto the pavement; during such overtaking, each vehicle honks several times (see [3]). (b) Sometimes the honk-recording, in one or both the recorders, gets mixed with human voice, police whistle or an overlapping honk, each of which has components in the 2-4KHz range. This changes f_1 or f_2 or both, giving erroneous high speed values. (c) The final possible reason is wrongly matched honks from two different vehicles, getting wrong f_1 or f_2.

4. There are a few *low* values of speed under free flow. One reason for this is that there is a natural tendency for vehicles to honk if they have to slow down for some reason, such as to warn a pedestrian crossing the road. That is, there is an inherent bias toward lower speeds in our speed sampling mechanism. Another reason is that, like in Fig. 10, some low speed estimates come from (badly-matched) honks outside the zone of interest.

Observations (3) and (4) essentially mean that there are some outlier speed values in our speed CDF. The next section (Sec. 6.4) shows how we can work around this.

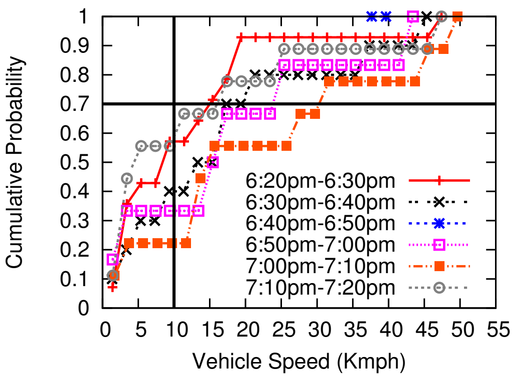

Figure 14: North-South Direction on Normal Day

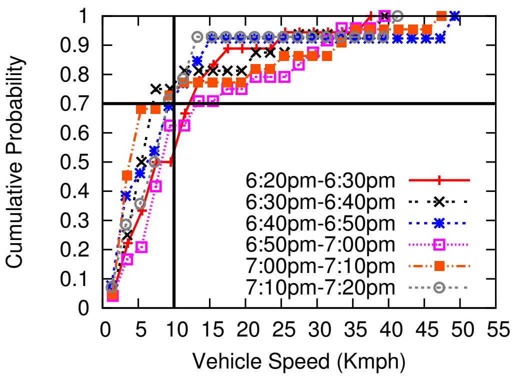

Figure 15: South-North Direction on Normal Day

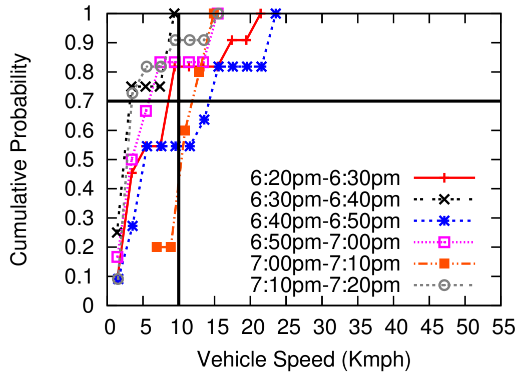

Figure 16: North-South Direction on Rainy Day

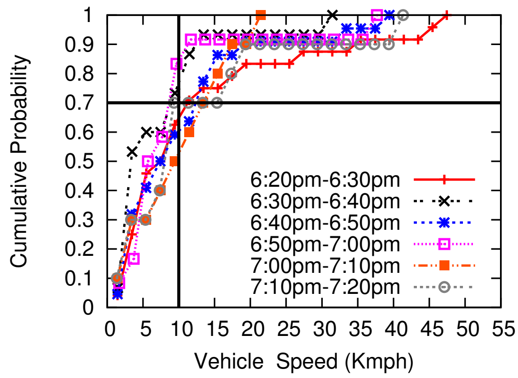

Figure 17: South-North Direction on Rainy Day

6.3 Direction sensitivity of speed estimates:

Our speed estimates are direction sensitive: each non-zero estimate is signed. The sign indicates whether the vehicle is moving from Recorder-1 to Recorder-2 or vice versa. Four hours of data collected in *Hira* was on a road which had traffic in both directions. The north-south direction always had free-flowing traffic, and during these four hours, the south-north direction was congested, due to queue build up prior to a congested intersection.

In such a scenario, we saw that our speed estimates were able to represent the two different traffic states, after removal of all the zero-speed estimates (which had ambiguity in the direction). A sample set of 6-plots for each direction is given in Fig. 14 and Fig. 15 respectively. The difference between the two sets of plots is apparent visually.

Rainy day: On the same road, a striking result is obtained from the data on 11^{th} Nov. There was unseasonal rain, due to a cyclone in the Arabian sea, and this made the traffic slow in both directions. This is clearly identified by our speed estimates, as seen from Fig. 16 and Fig. 17.

6.4 Metrics for traffic state classification

What metrics can we use to classify traffic state as congested versus free-flowing? The metric should be resilient to speed sample outliers like those in Fig. 12 & 13. We present two kinds of metrics: (a) speed-based and (b) non-speed based acoustic metrics.

6.4.1 Speed-based metrics:

From observing all our 10-min CDF plots, we arrive at the following two metrics: (1) 70^{th} *percentile speed* and (2) $P(v_s < 10Km/h)$. Both these metrics showed clear difference between the plots in congested and free-flowing traffic states. The visual difference can be readily seen in the plots of Fig. 12 versus Fig. 13. The 70th percentile horizontal line and the 10Km/h vertical line are given for visual aid.

We observed similar differences in all of our other CDF plots too. We summarize our data as follows. From each 10-min data, we get one sample of each of the above two metrics. The number of such samples obtained, their mean, and standard deviation, are given in Tab. 2.

	Hira		*Adi*	
Metric	Congested mean (s.d) [24 samples]	Free-flow mean (s.d) [54 samples]	Congested mean (s.d) [27 samples]	Free-flow mean (s.d) [27 samples]
70^{th} perc. speed (kmph)	12.2 (4.0)	18.2 (6.2)	7.7 (6.1)	21.1 (6.1)
Perc. speed < 10Kmph	65.6 (11.6)	51.1 (16.3)	79.5 (16.1)	37.6 (20.2)

Table 2: Speed based metrics

We can see a clear difference between congested and free-flowing states, for either road. The difference is much more stark for *Adi*, which is also what we observed visually.

6.4.2 Non-speed based acoustic metrics:

The several hours we spent by the road-side, collecting data, was tiring but gave us useful intuition about road noise. Congested traffic was inherently more noisy than free flowing: vehicles braking, engines revving, excessive honking, etc.. We now consider whether non-speed based acoustic metrics can be used to differentiate traffic states. We consider the following three metrics, computed over 10-min recording clips as earlier. (1) The number of honks detected. (2) The total duration of honks in 10-min (sum of durations of each honk detected). (3) And finally, the average noise level (across all frequencies), in dB.

	Hira		*Adi*	
Metric	Congested mean (s.d) [24 samples]	Free-flow mean (s.d) [30 samples]	Congested mean (s.d) [27 samples]	Free-flow mean (s.d) [27 samples]
Num. Honks	113 (30.4)	55.5 (21.1)	149.4 (27.8)	57.6 (21.2)
Honk duration (sec)	45.1 (12.4)	21.8 (9)	71.5 (21.4)	21.7 (9.2)
noise level (db)	-15.5(0.8)	-17.8(1.5)	-13.8(1.6)	-14.7(0.9)

Table 3: Non speed based acoustic metrics

Tab. 3 shows the mean across the various 10-min samples as well as the standard deviation, of the three metrics for the two roads under the two traffic states. All three metrics are averaged across recorders R1 and R2. For the first two metrics, we see that there is a clear difference between the values in congested versus free-flowing states. This is true for both *Hira* and *Adi*. For the third metric, the average noise level, although there is a difference, it is not as significant as in the other two non-speed metrics, especially in *Adi*.

6.5 Statistical divergence tests

For the above five metrics, is the difference between their values in congested versus free-flowing states statistically significant? To answer this, we employ two non parametric statistical hypothesis tests: the Mann-Whitney U test and the two sample Kolmogorov-Smirnov (KS) test. Non parametric tests are used to avoid assumptions about the underlying distributions of the metric samples.

For each of the metrics, we conjecture an appropriate *null hypothesis*. For instance, for the *70th percentile* metric, for *Hira*, we have the *null hypothesis* that the 24 samples from the congested state and 30 samples from the free-flowing state come from the same distribution. We thus have a total of twenty such hypotheses: five metrics x two roads x two statistical tests.

Tab. 4 lists the p-values from these 20 tests. We see that other than the noise metric in *Adi*, all p-values are very low. Thus the null hypotheses are rejected even at very low significance levels for these p-values.

For the noise level metric, for the *Adi* road, the null hypothesis is not rejected at the 0.001 significance level, but is rejected at the 0.01 significance level. This matches with our observation that the *Adi* road is noisy even in the free-flowing traffic state, due to several buses and large trucks.

	Mann-Whitney U test		Kolmogorov-Smirnov test	
Metric	*Hira*	*Adi*	*Hira*	*Adi*
70^{th} perc. Speed	2.00E-006	7.48E-007	6.16E-005	4.48E-004
Perc. Speed < 10 Kmph	1.05E-005	2.28E-004	3.57E-006	5.95E-004
Num. Honks	5.33E-015	2.13E-014	3.30E-014	5.36E-019
Honk duration	3.86E-014	3.89E-014	6.19E-014	6.53E-017
Noise	2.18E-013	0.0131	1.27E-014	0.0017

Table 4: p-values of statistical tests

6.6 Threshold based traffic state classification

Given the above high statistical difference, we propose a simple threshold-based traffic state classification, as follows. For a given metric, say *70th percentile speed*, we compute the mean value of this metric across all congested 10-min windows. Denote it as, say X_{cong}. Similarly we compute the mean across all 10-min windows marked as free-flowing, and denote it as X_{free}. For the data we have collected, X_{cong} and X_{free} are given in Tab. 2 & 3 for the 5 metrics.

We take the threshold for traffic state classification based on that metric as $X_{thr} = (X_{cong} + X_{free})/2$. For instance, for the *70th percentile speed* metric, $X_{thr} = (7.7 + 21.1)/2 = 14.4Km/h$ for *Adi*. Essentially, we have trained the classification algorithm using our data set, and any further 10-min data would be classified as congested versus free-flowing based on this threshold. For the *70th percentile speed* metric, if a future 10-min measurement has a metric value > 14.4 Km/h, it would be classified as free-flowing, and as congested otherwise.

The various metric mean values, as seen from Tab. 2 & 3, are different for the different roads. So the thresholds we calculate should be road specific.

How effective is this threshold-based classification? To determine this, we have used the following method. For each experimental 10-min run, marked with ground truth (congested versus free-flowing) in our data, we seek to classify it using the above threshold-based mechanism. The threshold itself is determined using all the data on that road, except that 10-min run itself. If our classification detects congestion for that 10-min window, whereas the ground-truth is marked as free-flowing, this constitutes a false positive in congestion detection. The vice-versa case is a false-negative.

	Hira		*Adi*	
Metric	Fp (%)	Fn (%)	Fp (%)	Fn (%)
70th perc. Speed	24.1	8.3	12.1	5.6
Perc. Speed < 10Kmph	20.9	25.3	27.2	18.3
Num. Honks	10.7	17.4	0.0	5.9
Honk duration	7.1	19.6	0.0	5.9
Noise	19.6	6.5	74.3	65.6

Table 5: Threshold based congestion detection

Computing across all 10-min samples, we can thus calculate the false-positive and false-negative rate, for our traffic congestion detection mechanism. Tab. 5 summarizes the false-positive and false-negative rates for the various metrics, on the two roads.

We see that we achieve reasonably good accuracy; in most cases, the false positive and false negative rates are under 20%, and in many cases under 10%. We believe that such accuracy is significant, especially given that the ground-truth labeling was just by visual observation. The current state-of-the-art in widespread use is highly coarse-grained radio announcements for traffic updates. We believe that our classification mechanism will provide similar or better updates automatedly.

In Tab. 5, the classification accuracy for the metrics based on number and duration of honks, is especially good on *Adi*. However, as noted earlier, non-speed based metrics are direction insensitive.

We believe that there is scope for various metrics to be used in conjunction with one another to better decide traffic state. For example, during free flow, it might happen that, there are few honk samples from fast moving vehicles, due to which the speed based metrics give a pessimistic view of the traffic. But if the total duration of honks is considered, the classification could be more accurate.

One final point we note from Tab. 5 is that the metric choice could itself be specific to a road stretch; for instance, noise level is a more useful metric at $Hira$ than at Adi.

6.7 Detecting the onset of congestion

We present one final experiment to show that our technique can detect the onset of congestion. For this, we present data from a continuous two-hour recording, 6pm-8pm, on 4^{th} Dec, 2009, on *Adi*. The traffic state is initially free flowing. It starts becoming congested from about 6.35pm. Heavy congestion set in by 7.10pm. The values of the four metrics (1) Number of honks, (2) Duration of honks in secs, (3) 70^{th} percentile speed and (4) Percentile of speeds $<$ 10 Km/h are plotted in Fig. 18. There are 12 values for each metric, corresponding to 12 clips of 10 mins, over 2 hours.

The figures also show as horizontal lines, the classification thresholds computed, as per Sec. 6.6. For this, we use all other data on *Adi*, except these two hours, as training set. The plots in Fig. 18 show that according to each metric, we start in free-flow state, and finally move to congested state in the 2-hour duration. The four metrics *70th percentile speed*,

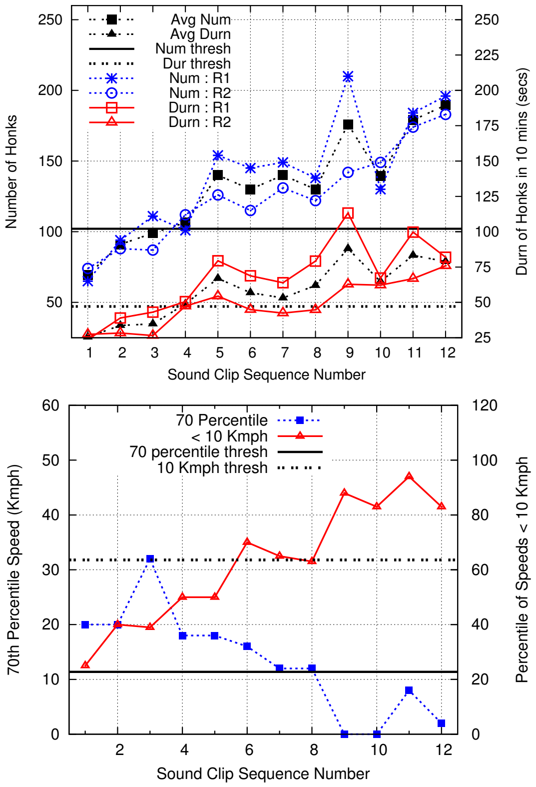

Figure 18: Change in metric values in two hours

$P(v_s < 10Km/h)$, *number of honks*, and *duration of honks* detect congestion at clip numbers 9, 6, 4, and 4 respectively.

The number and duration of honks show an early increase because, even as congestion is setting in, traffic becomes more chaotic. Thus there is a state where vehicles are moving yet honking more due to the increasing disorder. Even though the four metrics do not agree on the classification when the traffic congestion is setting in, they all finally report congestion.

The first plot also shows the number and duration of honks at recorders R1 and R2 separately. We plot this to show that R1 consistently shows more number and duration of honks compared to R2. This supports our earlier observation that vehicle honks are directional, with bias toward the direction of motion.

We make a final observation using the above data; which supports our earlier conjecture that metrics used in conjunction with one another provide more information than using them individually. Clip number 10 shows relatively fewer honks and lower honk duration, as compared to other clips in congestion. But a look at the speed-based metrics for this clip tells that the 70^{th} percentile speed is 0 Km/h, and 80% of the speeds are $<$ 10 Km/h. Thus the clip clearly belongs to congested state. Such use of metrics in conjunction with one another is part of our future work.

7. DISCUSSION AND CONCLUSION

7.1 Practical difficulties:

We faced several practical issues in the course of the experiments. With respect to the use of the phones for recordings, we found during our initial several weeks of experiments that the phones used to go out of synchrony for several honks. We conjectured various possible causes for this: echoes, other interfering applications in the N79 phones, the WiFi or Bluetooth interface activity, on-phone GPS etc. The behavior was sporadic and non-repeatable. After a lot of heart-burn, we finally diagnosed the problem to be as innocuous as button-presses on the phone! If the user pressed a button to light-up the sleeping display, just to see if all was well, it caused a large delay (1-2 sec) in the recording.

In the aftermath of the terrorist strikes in Mumbai, our activities using laptops, phones, external microphones, etc. aroused a lot of suspicion. While this caused procedural delay for us, it was nice to see that people were vigilant! They were also quite helpful once we showed our credentials and explained our goals.

7.2 Future work:

(1) Going forward, we are working on a hardware prototype which can be installed at several locations for data collection. (2) Some aspects of our algorithm can be further enhanced, such as filtering the spurious honk matches. (3) The threshold based classification is naive, but effective; more powerful SVM classifiers can be designed. (4) A finer grained traffic state classification, (5) using consecutive sensor pairs to estimate traffic queue length, and (6) the use of traffic state information over several roads for providing travel time estimates, are other interesting aspects.

7.3 Conclusion:

In conclusion, this paper has considered the important problem of providing dynamic information about road traffic to users on the move. Our technique is focused on *chaotic* traffic conditions. We develop an *inexpensive* mechanism for vehicle speed estimation using the Doppler shift of honks from moving vehicles. We use two sensors (audio recorders). The speed estimation consists of three steps: honk detection, honk matching, and frequency extraction. Based on extensive experiments, we have presented and evaluated algorithms for each of these three steps. We design five different metrics: 70th percentile speed, $P(v_s < 10Km/h)$, number of honks, duration of honks, and the average noise level, to classify traffic state as congested versus free-flowing. MWU and two sample KS tests on these five metrics show statistical divergence at the 0.1% significance level for two different city roads. Thus a threshold based traffic state classification is straightforward: our results show that such classification matches with ground truth 75-100% of the time. This indicates promise for widespread deployment of our technique.

Acknowledgment

This work is supported in part by Microsoft Research India. We thank Prasad Gokhale, Prof. Om Damani, Amit Srivastava, Ajinkya Joshi, Piyali Dey, Akash Sharma, Vijay Gabale and Lokendra Kumar Singh, who helped with various road experiments. Zahir Koradia and Prof. Preeti Rao provided valuable input at the inception of this project. Thanks are also due to all those who honked, and those pedestrians who jay-walked to cause many of the honks!

8. REFERENCES

[1] Rijurekha Sen, Vishal Sevani, Prashima Sharma, Zahir Koradia, and Bhaskaran Raman. Challenges In Communication Assisted Road Transportation Systems for Developing Regions. In *NSDR'09*, Oct 2009.

[2] `http://en.wikipedia.org/wiki/Infrastructure_in_Bangalore`.

[3] Chaotic traffic: representative videos. `http://www.cse.iitb.ac.in/~br/horn-ok-please/`.

[4] Video of a road intersection in bangalore, india. `http://research.microsoft.com/en-us/um/india/groups/mns/projects/nericell/index.htm`.

[5] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee. Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones. In *SenSys*, Nov 2008.

[6] Jungkeun Yoon, Brian Noble, and Mingyan Liu. Surface street traffic estimation. In *Mobisys*, 2007.

[7] Shalini Singh. India adds 83 mn mobile users in a year. `http://timesofindia.indiatimes.com/Business/India_Business/India_adds_83_mn_mobile_users_in_a_year/articleshow/2786690.cms`, Feb 2008.

[8] Harsimran Singh. Mobile boom helps India reach internet goal before time. `http://economictimes.indiatimes.com/articleshow/2341785.cms`, Sep 2007.

[9] Horn Ok Please: Wikipedia. `http://en.wikipedia.org/wiki/Horn_OK_Please`.

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

[11] Li Li, Long Chen, Xiaofei Huang, and Jian Huang. A Traffic Congestion Estimation Approach from Video Using Time-Spatial Imagery. In *ICINIS '08, First International Conference on Intelligent Networks and Intelligent Systems*.

[12] Sing Y. Cheung, Sinem Coleri, Baris Dundar, Sumitra Ganesh, Chin-Woo Tan, and Pravin Varaiya. Traffic measurement and vehicle classification with a single magnetic sensor. Paper ucb-its-pwp-2004-7, California Partners for Advanced Transit and Highways (PATH), 2004.

[13] Volkan Cevher, Rama Chellappa, and James H. McClellan. Vehicle speed estimation using acoustic wave patterns. *IEEE Transactions on Signal Processing*, pages 30–47, 2009.

[14] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring. In *MobiSys'08*, Jun 2008.

[15] Bhaskaran Raman, Kameswari Chebrolu, Naveen Madabhushi, Dattatraya Y Gokhale, Phani K Valiveti, and Dheeraj Jain. Implications of Link Range and (In)Stability on Sensor Network Architecture. In *WiNTECH*, Sep 2006.

[16] Fast Fourier Transform. `http://en.wikipedia.org/wiki/Fast_Fourier_transform`.