Acoustic Sensor Based Road Congestion Detection in Developing Regions

Rijurekha Sen
under the guidance of
Prof. Bhaskaran Raman

Abstract

Road congestion is a common problem worldwide. Intelligent Transport Systems (ITS), in developed countries, seek to alleviate this problem using technology. But such ITS are mostly inapplicable in developing regions due to high cost and assumptions of orderly traffic. Efforts in developing regions have been few. We seek to develop low-cost ITS techniques to detect road congestion which will work in disorderly road conditions. We propose techniques based on road noise, especially vehicular honks, a characteristic feature of Indian roads. We take Indian traffic conditions as an example for our analysis but we believe that most of our claims and experimental results can be extended to other developing countries too.

Our system comprises a pair of road side acoustic sensors, separated by a known distance. If a 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.

We have done extensive experiments in semi-controlled settings as well as three different real road scenarios under different traffic conditions. From the road-side recordings, we have identified five possible metrics to characterize traffic state: 70th percentile of vehicle speed, percentile of speeds below 10 Kmph, number of honks, duration of honks and noise level. Statistical divergence of congested and free flowing traffic data, based on these five metrics, is verified at 99% confidence using the two sample KS and MWU tests. We have designed a threshold based technique to classify traffic state into congested and free flowing. n-fold cross validation of this technique gives minimum classification accuracy of 75% for all five metrics.

1 Introduction

1.1 Need for ITS in India

Road traffic conditions in India have particularly worsened in recent times as is evident from one of the statistics which states that the average number of vehicles on Indian roads is growing at an enormous rate of 10.16 percent annually since the last five years [1]. The condition is particularly alarming in metropolitan cities like Mumbai, where vehicle penetration has reached over 590 vehicles per km of road stretch and in Bangalore, where about 5 million vehicles ply on a road network that extends barely 3000 kms [2, 3]. This is leading to higher levels of road congestion, longer and unpredictable travel times, wastage of time and fuel for commuters and more cases of road accidents. Growth in infrastructure has been slow due to various reasons such as high cost, lack of space, bureaucracy, etc. It thus needs to be supplemented with ITS techniques that utilize existing infrastructure more efficiently to give better traffic management. What is the congestion level at important road intersections? Given a source and a destination, what is the route that will take least travel time? How should new infrastructure such as flyovers, freeways, etc be planned to minimize congestion? There is an evident need of ITS based applications that can answer such questions to make traveling on Indian roads less cumbersome.

1.2 Inapplicability of Existing ITS Techniques in India

Many ITS applications have already been designed, implemented, deployed and are being used in developed countries. But there are some major differences between the road and traffic conditions that are prevalent there and in India. For e.g. in USA, freeways and expressways extend to over 75,000 kms [4], while [1] claims that only about 200 kms of expressways are present in India. The nature of traffic is fundamentally different from that in the developed traffic. The difference needs to be experienced to be fully understood, but an appreciation can be partially gleaned from the representative video at [5, 6]. Unlike traffic in developed countries, traffic on Indian city-roads 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 [7]. Roads are also generally not as well maintained, with potholes being common. Thus it is intuitive that the various techniques that have been developed in the context of traffic conditions in developed countries will not be applicable directly in an Indian context. We elaborate more on this intuition in Section 2.

1.3 Our Proposed ITS Technique

The chaos on Indian roads cause traffic to be inherently noisy. A characteristic feature of this traffic noise is excessive use of *honks*. Honks are an integral part of driving on Indian roads; cautioning pedestrians, alerting fellow drivers – everything is expressed through honking. This gives rise to a system of moving sources (vehicles) of sound (honks). Speed of such vehicles can be estimated using Doppler shift of the honk frequency. Our proposed ITS technique uses this concept.

1.4 Thesis Goal

We envision a system where cheap acoustic sensors i.e. recorders are placed on the roadside and vehicle speeds are estimated from the Doppler shift of vehicular honks recorded by the sensors. Till date, we have designed a two sensor architecture to achieve this and designed algorithms for honk detection, honk matching across the two sensors and frequency extraction for speed estimation. Our speed estimates, validated through experiments on university campus roads and a pair of city roads show error under 6 Kmph. This level of accuracy is enough to use the speed estimates in binary classification of traffic states into congested and freeflowing. We have designed five metrics, based on whose values, we can do such binary classification. Two of the metrics are speed based, namely 70^{th} percentile of speed and percentile of speed below 10 Kmph. The remaining three are non speed based acoustic metrics - number of honks, duration of honks and noise level. Statistical divergence of congested vs freeflowing traffic data has been verified at 99% confidence level using MWU and two sample KS tests. We have developed a threshold based technique for binary classification of traffic data into congested and freeflowing states. Using 18 hours of city road data for both training and testing and n-fold cross validation, we have achieved minimum classification accuracy of 75% for each metric.

1.5 My Contributions

Prof. Bhaskar Raman had the idea of using Doppler shift of honk frequency in vehicle speed estimation. Zahir Koradia did the initial implementation and some semi-controlled experiments for proof of concept of this technique. The details of Zahir's work can be found in [8]. I have moved forward the initial work through the following steps.

- Extensive study of existing ITS techniques and their applicability in India and other developing regions. This was done as part of my seminar. An integration of this seminar along with those of Vishal Sevani and Prashima Sharma can be found in [8].

- Empirical data collection from real roads to see if sufficient vehicles honk in practice and what are the common properties of honk like average duration and frequency range. Details of this step can be found in Section 5.

- Implementing a sound based synchronization technique to time synchronize the recorder pairs. Details of this step can be found in Section 4.5.

- Speed estimation using Doppler shift of honks has three components - detecting honks in each sensor's recording, matching a honk as the same across the two sensors, extracting appropriate frequencies from matched honks to estimate speed. While Zahir had the basic implementation of all these, his work was incomplete. The algorithms had many manual aspects and most parameters were arbitrarily chosen. I refined the algorithms manifold through in-depth study, semi-controlled experiments and data analysis. Prashima Sharma has worked with me throughout this step, the division of work being 60% by me and 40% by her. Details of this step can be found in Sections 6, 7, 8.

- Evaluation of the speed estimation technique using semi-controlled experiments, both inside campus and on real roads. The in-campus experiments and data analysis were done along with Prashima Sharma. Details of this step can be found in Section 9.

- Studying the application of the technique in traffic state classification on real roads through experiments. This involved 18 hours of data collection sitting on the road, data analysis, design of proper metrics to do traffic state classification, verifying statistical divergence of data of congested vs freeflow based on these metrics, implementing a simple threshold based technique to do a two-way traffic state classification and validating the last technique using n-fold cross validation. Details of this step can be found in Section 10.

2 Related Work

ITS can be developed for many applications - monitoring the road surface condition, predicting the arrival time of public transport, estimating travel time between two places, automatic toll collection, detection of road congestion to redirect commuters to less congested roads and so on. This work is on developing acoustic sensor based techniques for road congestion detection in developing regions. So we focus specifically on existing road congestion detection techniques and seek to gauge what advantages or disadvantages existing approaches have in our context.

ITS techniques can be divided into two broad categories - *fixed sensor based* where the sensors, that gather various road related information, are statically placed on or by the side of the road. The second category is *probe vehicle based*, where the sensors are mobile and placed in a subset of vehicles that ply the road.

2.1 Existing fixed sensing techniques

There are three main on-road sensing techniques as follows.

2.1.1 Dual loop detectors

Technique Pairs of inductive loop detectors can be used to identify vehicles based on their length [9]. Identifying the same vehicle at the two detectors can give an estimate of travel time between the two detectors. Deviation from expected travel time can signal congestion.

Critique Widespread application of the technique in congestion detection can be prohibitive in terms of infrastructure cost, as dual loop detectors need to be constructed at regular intervals along the road. As given in [10], a vehicle loop detector costs $700 for a loop, $2500 for a controller, $5000 for a controller cabinet, $300000 for fiber optic cable per mile and 10% of the original installation cost for annual maintenance as of 1999. Furthermore, the inherent assumption of lane-based orderly traffic makes it inapplicable for chaotic road conditions.

2.1.2 Image sensors

Technique Some papers [11] use image sensors, deployed on road side and measure congestion level by image processing techniques, where slower the images change with time, higher is the level of congestion.

Critique These techniques have high installation and maintenance cost, [12, 8], running into $10-20K per installation.

2.1.3 Magnetic sensors

Technique [13] uses a single magnetic sensor to detect the ontime of vehicles pasing it, calculates the length of vehicle based on the ontime assuming constant speed common to all vehicles. From a large number of samples of ontime and vehicle length, they calculate median of the two metrics and estimate median speed as $\frac{medianlength}{medianontime}$. If this median speed deviates from expected median speed, congestion is reported.

Critique While these techniques can be relatively inexpensive, they also make assumptions of traffic orderliness [8]. Secondly, detection of motorcycles is unreliable [13], which form a substantial part of road traffic in developing regions. Thirdly, the assumption of low variability in vehicle speeds also does not hold in developing regions where heavy slow moving trucks and high speed motorbikes ply the same road.

2.2 Existing Probe-based Techniques

Techniques In probe based techniques, the main focus is on GPS based sensing. A lot of work is being done in developed countries to devise, implement and deploy such systems. A small amount of work has been done to use sensors available on smartphones. The basic details of these works are discussed below.

- [12] 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.

  Critique - Segments are bounded by signalized intersections. In India, even within such a segment, traffic conditions will vary as there will many intermediate intersections, not signalized, where drivers will follow random protocols to decide who will go first.

- The Mobile Millennium project of UCBerkeley [14] includes a six month pilot deployment of GPS technology, where thousands of GPS mobile phones were placed in a subset of vehicle within a focus area. Participating users agreed to place these cell phones in their vehicles in order to transmit positioning data. The phones received live traffic information from a map application for free. In context of this deployment, they have developed algorithms for travel time estimation, optimal sensor placement and protecting user privacy.
- [15] attempts to predict bus travel times in Chennai, a metropolitan city of India. The authors devise a system with GPS data and linear regression techniques using bus dwell times at intersections and bus stops, number of passengers, average speed of bus, lengths of 2 lane road, 4 lane road and 6 lane road between start and target bus stops.

  Critique - There are three open issues with this system– (a) deviation of the system from linear regression model in case of road incidents, (b) infeasibility of collecting passenger data manually in case of practical deployments and (c) recalibrating the linear model for each different bus route.

- [16] develops techniques to augment sparse GPS data with Wi-fi localization information from urban hotspots. Secondly it develops a new technique to map GPS traces to road segments using hidden Markov models with Viterbi matching. Thirdly they experimentally quantify power consumption in mobile devices if GPS sensing is kept on.
- [17] tries to characterize traffic and road conditions in the Indian city of Bangalore at low infrastructure cost using mobile phones that have several sensing components like microphone, GPS

receiver, accelerometer and camera. The main contribution of their work is automatic reorientation of accelerometer. They devise a triggered sensing mechanism to save power on resource constrained mobile devices. They roughly localize using GSM, radio for which consumes low power. If this rough localization shows the phone to be in region of interest, say an impending road intersection, GPS receiver is turned on to localize more accurately. Accelerometer is next turned on to detect braking and if there is sufficient braking, microphone is turned on to detect honks. If honks also are substantial, congestion is reported.

Critique - Accuracy and cost effectiveness of this technique for road monitoring is yet to be judged on the basis of how these mobile phone based sensors will be deployed (scale and density), how data from them will be aggregated, processed and used to infer road conditions.

Issues with deploying probe based techniques in India:

Though probe based techniques are inexpensive and do not assume traffic to be orderly, there are some general issues in their deployment in the current Indian traffic scenario for the following reasons.

- In India, proliferation of GPS receivers in vehicles is quite low. Few fleet companies like Meru [18], Easycabs [19] and state transport companies like BMTC [20] have GPS units installed in their devices. The first two fleet companies have services only in Mumbai, Hyderabad, Bangalore and Delhi and have around 5000 vehicles in each city. BMTC has round 5500 vehicles in Bangalore. But there are 42 cities in India with population exceeding 1 million [21] where there is no such service. Even for the cities which do have fleet services, percentage of fleet vehicles is low compared to the total vehicle number. Low percentage of probes might cause sensed information to be outliers and not representative of the general traffic trend.
- The fleet companies might have GPS information about their vehicles, but since that location information is the key to their business success, there is little reason for them to share travel time information with other transport companies and the common commuters.
- Smartphone penetration in India is also quite low, is about only 2% as claimed by [22], though mobile phone penetration is extremely high. Most people have low end phones unable to take part in participatory sensing.
- Even for the people who do have smartphones, it is very difficult to think of a business model to attract them to take part in participatory sensing as that involves sensing as well as communication costs. Power drainage figures, as reported in [16] to keep GPS receivers in on mode is quite high.

Inspite of the above issues, probe based techniques are definitely more promising than their fixed sensing counterpart due to low cost and lack of unrealistic assumptions. If coverage of any city by probe vehicles is high, practical ITS techniques can be designed and deployed.

2.3 Existing Audio based Techniques

Technique Vehicle speed estimation using Doppler shift of frequencies of vehicular sound is a well known idea. 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.

Critique 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.

Technique [23] suggests a low cost technique of vehicle speed estimation by classifying acoustic wave patterns, recorded with a single roadside acoustic sensor. It uses engine, tire, exhaust and air turbulence noise as vehicular sound, Doppler shift of whose frequencies is used to compute vehicular speed.

Critique Presence of highly noisy traffic and a huge variety of acoustic signatures of vehicles will limit the applicability of such techniques on Indian roads.

3 Our Approach: Vehicle Speed Estimation Using Doppler Shift of Honks

Noise in Indian roads has a unique characteristic, namely *abundance of vehicular honks*. Honks are present, in India, under all road conditions: congested or otherwise. They are tightly knit with the driving "protocol", so much so that honking is considered an aspect of "safe" driving. And such thinking is not without truth since, in many situations, honks are expected by drivers/pedestrians to avoid accidents. We seek to develop a vehicle speed estimation technique, using road-side acoustic sensors that record vehicular honks. Based on the Doppler shift observed in the measured frequencies of the honks, it is possible to estimate the speed with which the honking vehicle is moving.

If source of sound moves with speed v_s, receiver of sound is stationary, emitted frequency of sound is f_0 and speed of sound is v, then

- when source moving away from the receiver, frequency observed at receiver is given by,

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

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

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

If f_0 is known, v_s can be estimated easily from equations 1 and 2. But it is not easy to guess f_0 as different honks have different base frequencies. In absence of f_0, we thus need to use an alternate arrangement using two sound receivers, as shown in Fig.1. When a moving vehicle blows honk in between the two receivers, it is approaching one receiver and receding from the other. Substituting value of f_0 from equation 1 in equation 2, we get following equation,

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

Estimation of v_s is thus a three step process.

Detection of honk from the recording at each of the two sensors in presence of background noise.

Matching honks across recordings in two sensors to identify the same honks recorded in both.

Extracting frequencies $f1$ **and** $f2$ in Equation 3 from each honk pair, matched as the same across the two sensors.

Using the $f1$, $f2$ obtained through the above three step process and taking speed of sound (v) = 340 m/sec, we can estimate v_s applying Equation 3.

The system we envision comprises of inexpensive road-side acoustic sensors, collecting dynamic information about vehicle movement on the roads, using the technique discussed above. 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.

The advantages of our approach are listed below.

- The wireless-enabled sensors are cheap. The components needed are a recorder, a processor, a flash memory, a local radio, a clock and some connectivity like GPRS/GSM. It is very similar to the mobile phone hardware, which one can get as cheap as $20 in India.

- The technique is custom made for chaotic traffic. There is no orderliness assumption, in fact, more the chaos, more is the number of vehicles honking and higher is the number of speed estimates obtained.

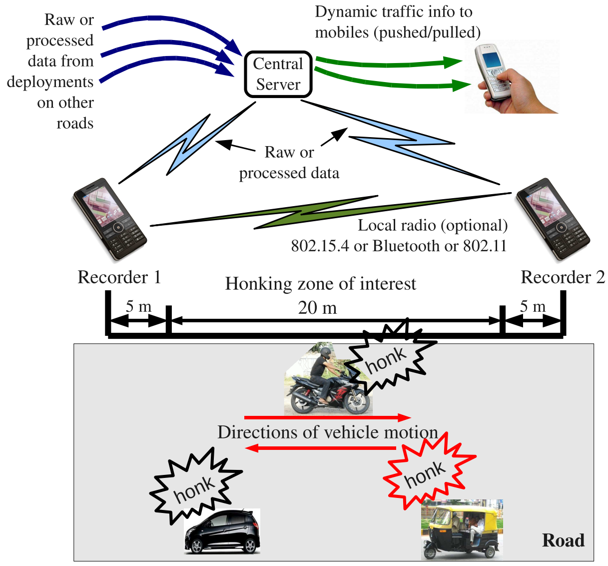

Figure 1. System Architecture

- 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.

- 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.). The lack of penetration of GPS receivers/ smartphones in India, power drainage and privacy issues and lack of incentives in participatory sensing have no role to play here. Most vehicles honk, when traffic is chaotic and congested.

Scope of this work - Till date, we have done some empirical analysis of road noise data to learn honk properties. We have designed and implemented algorithms for the three step speed estimation. Road experiments to evaluate our speed estimation algorithm and verify its applicability in congestion detection has been done. Some specific things we have **not done yet** are as follows.

- We have used two N79 phones as recorders, and not yet designed any acoustic sensor customized for our purpose.

- All analysis has been offline, there has been no connectivity or data transfer from the recorders in the experimental site to any central server.

- Recorders have been positioned in places showing congestion in peak hours and free flow otherwise. Places were also chosen according to convenience of standing/sitting with the recorder. Optimal sensor placement based on traffic criticality or road coverage has not been studied yet.

- The data analysis has been strictly in terms of detecting patterns and designing metrics for differentiating congested traffic data from that of freeflowing. Correlating data from different sensor pairs to estimate travel time, correlating data from consecutive sensor pairs to estimate vehicle queue length, doing time series analysis to detect congestion patterns at specific times of the day, forecasting future traffic state from present state using machine learning techniques on historical data have not been done.

- No user level application has been designed to provide information to commuters on the road.

All the above things, not done yet, provide interesting avenues of future work.

4 Details of Experimental Methodology

There are three distinct phases of the work done till now- a) empirical study of road noise, b) designing and implementing algorithms for speed estimation and c) designing metrics to do traffic state classification and verifying its applicability on real road data.

In the empirical study of road noise, we seek to know what is the average honk length and what is the peak frequency range of a honk, so that we can use this knowledge in our algorithm design. We also seek to learn how often vehicles honk as that will decide the feasibility of practical deployment of our system.

In the algorithm design phase, our main questions are as follows: how can we detect honks in presence of substantial background noise of roads? How can we match honks recorded at the two sensors as one? How can we extract $f1$ and $f2$ in Equation 3 to estimate speed? How accurate are our speed estimates thus obtained?

In the third phase, we ask questions like are the speed estimates obtained from real city roads representative of the contemporary traffic state? Can we distinguish traffic states as congested or freeflowing using some speed-based metrics? Can any non-speed based acoustic metric be used in conjunction with speed-based metrics for traffic state classification? Is there a way to detect traffic state for individual traffic directions on a bidirectional road? Will data for congested and free-flowing traffic be statistically different based on the metrics that we design? Is there any way to classify new traffic data into congested or freeflowing state based on historical data and if so, what is the classification accuracy? Can we detect onset of congestion i.e. the transition from freeflowing to congested state?

Extensive experiments, both on campus roads and city roads have been done to find answers to the questions above. The setups, hardware and software used in the experiments are detailed in this section.

4.1 Experimental setups

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 10 m wide overall. We also have 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 [6]. Both *Hira* and *Adi* are known for their congestion at peak times, the latter more so than the former.

4.2 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) If the inter-sensor distance is large with respect to the road width, the inaccuracy is low. Most city roads are at most two "lanes" wide, or about 5m each way. So we expect the speed estimation inaccuracy due to this to be small. (2) Further, 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, to reduce the inaccuracy.

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

4.3 Sensor placement

There are several issues related to how far the sensors are to be placed. The two sensors need to be sufficiently 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 in future, 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 metres if not more, even for the relatively high frequency of 2.4GHz [24] for 802.15.4 or Bluetooth or 802.11.

4.4 Hardware and Software Used

Our envisioned deployment will have custom-built hardware for recording, which we have not worked on till now. So we use the inbuilt voice recorder utility of Nokia N79 phones for recording. Three open source softwares –*mp4tomp3converter v3.0*, *Praat* and *Audacity* are used for manual preprocessing and analysis of the recordings. Though these are fairly complex audio processing softwares having a wide array of functionalities, we use a limited set of operations as follows.

- Voice recorder utility in N79 saves the recordings in MP4 format. We convert MP4 to MP3 through *mp4tomp3converter v3.0* converter.
- The recording is downsampled to 16 Khz from 48 Khz, converted to mono channel from stereo channel and saved as a 16 bit encoded wav file using *Audacity*. The wav format is easy for processing through open source libraries like Libsndfile. According to [17], frequency range of honks is within 2-4 Khz, so 8 Khz sampling frequency would be enough according to Nyquist's theorem. We double the sampling frequency to reduce noise. Stereo channel and higher bit encoding do not add any benefit to our analysis, so they are not used.
- *Praat* shows spectrogram of the recordings with time on the x-axis and frequency on the y-axis with higher amplitudes colored darker. We can *see* sound peaks in the recording as dark bands as well as *hear* them by playing. Thus *Praat* provides a two way identification of characteristic sounds.
- The spectrogram of a characteristic sound in *Praat*, see Fig.2, also shows the start time and the end time in microsecs level granularity.
- As seen in Fig.3, *Audacity* can be used to see which frequencies have the highest amplitudes in the recordings.
- Bandpassing is done using *Audacity* to reduce amplitude of sound outside frequency of interest.

4.5 Time Synchronization of Sensor Pair

Time synchronization of the audio files recorded in the two recorders is an important subproblem for the following reason. If Recorder1 starts at time t_0 and Recorder2 starts at time t_1 according to some global clock and $t_1 - t_0 = \Delta t$, then any event at time t according to the global clock in Recording2, should be matched a) not with an event at time t according to global clock in Recording1, but b) with an event at time $t+\Delta t$. If we erroneously do a) instead of b), we might have very bad consequences in our case where honks from two different vehicles might get matched as the same. The basic problem is Recorder1 has Δt time of extra recording in the beginning that needs to be clipped. This is a precursor for any further experiment or algorithm development. Hence we give the details of this now.

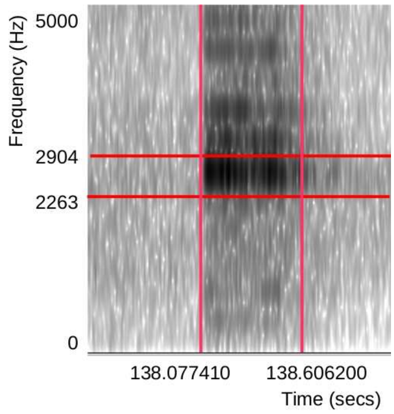

Figure 2. Spectrogram in *Praat*

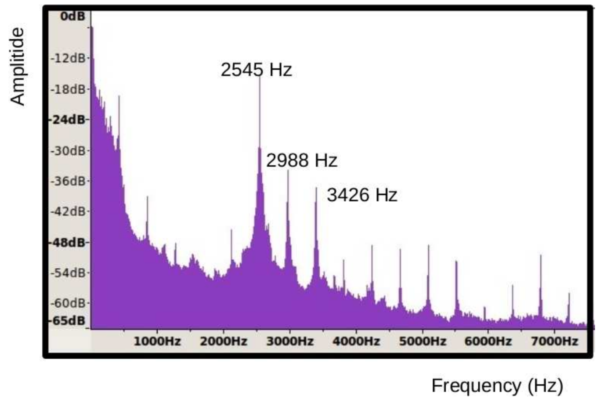

Figure 3. Spectrum in *Audacity*

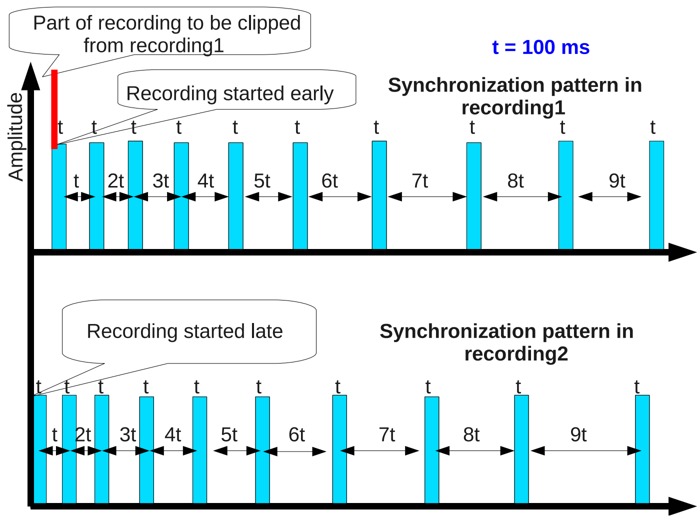

Figure 4. Time synchronizing audio files

4.5.1 Synchronization Method

In practice, it is not possible to know which recorder started early, let alone knowing the precise value of Δt. We thus take the idea of node synchronization using wireless messages that the nodes receive at the same time and start counting $time = 0$ from then. We seek to record a common sound in the two recorders so that the two recorders can start counting $time = 0$ from the time they record the common sound. The common sound to be recorded needs to have two properties i) it should have a distinct pattern to be distinguished as a special sound ii) the starting needs to be easily detectable, so that $time = 0$ in the two recordings can be assigned to the two detected start times in the two recordings and they should be within microseconds of each other.

For the common sound, we synthetically generate a wav file using *Matlab* that has 10 square waves, each of 250 Hz frequency and 100 millisecs duration, separated from each other by zeros whose durations increase from 100 millisecs to 900 millisecs in steps of 100 millisecs (See Fig.4). When we start Recorder1 and Recorder2, we initially play this wav file. It gets recorded in the two recorders which then continue to do the recording of road data in the same audio file. At the end of recording, we detect the start of the common sound, $t1$ and $t2$ in the two files respectively, $t2$ being greater than $t1$ and clip $t2 - t1$ part of the second file from the beginning.

The common sound has a self non-repetitive pattern that makes each crest-trough pair in the pattern unique. There is no chance of matching one crest-trough pair in a recording with another crest-trough pair in the other recording, thus reducing chances of synchronization error. Secondly, even if one or both recorders start so late that this common sound has already started playing, still we can clearly identify the crest-trough pair that the recorder starts to record as each pair is unique. This allows us the late start of the recorders, though the amount of delay after the common sound starts playing should be less than the duration of the common sound before the last crest-trough pair i.e. 5.5 ms.

We use this audio based approach for synchronization because we have to record anyway. So using other synchronization mechanisms like Wi-Fi, Bluetooth or GPS based techniques, each of which is available on the N-79, will need extra effort with no added advantage. In actual deployments, with automated data collections, no common sound playing mechanism will be there. Then we will need to design a different synchronization mechanism, maybe based on local radio like Wi-Fi, Bluetooth or Zigbee.

4.5.2 Synchronization Error

The error in synchronization is the error in detecting the start of the common sound. We seek to quantify this error in the following way. The problem of detecting the start of the common sound is same as the problem of detecting the start of any of the 10 square waves. We know the expected difference between the start times of two consecutive square waves. We take the absolute value of the difference between this expected value and the calculated value as error (See Fig.5).

We do the above procedure for 70 pairs of square waves in eight different recordings. The results are shown in Fig.6. The maximum error is of concern here which is 62 or 63 microsecs. Majority of the errors are zeros.

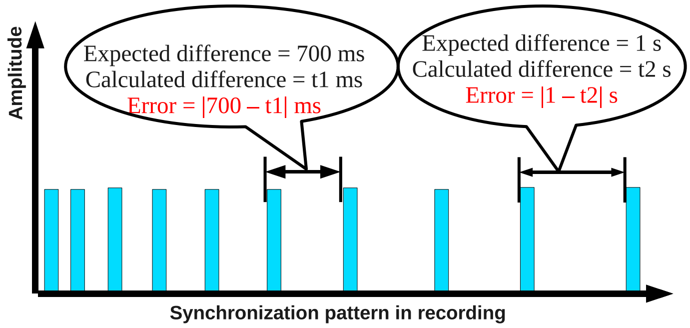

Figure 5. Method to calculate synchronization error

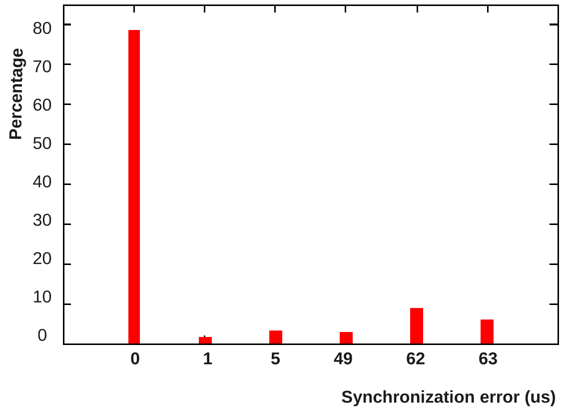

Figure 6. Synchronization error

5 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) what frequencies have highest amplitudes in a honk?

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 in *Praat*. Such dark regions indicate a sound peak. 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.

5.1 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.

5.2 How long do vehicles honk?

Fig. 7 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.

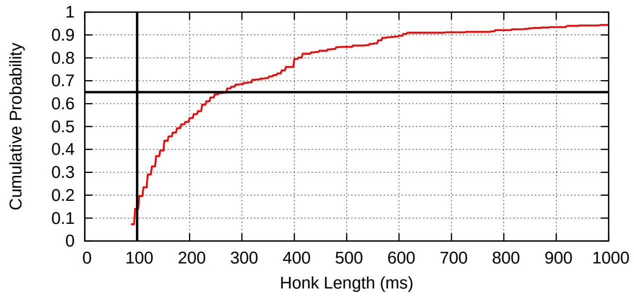

Figure 7. CDF of honk length

5.3 What is the audio frequency of honks?

We use the Discrete Fast Fourier Transform (FFT) [25] tool in the *Audacity* software to determine the honks' dominant audio frequencies. Nericell [7] 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.

6 Honk Detection

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

Nericell [7] uses the following simple honk detection algorithm. The recording is broken up into 100ms windows. A Discrete Fast Fourier Transform (FFT) [25] 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.

Band-pass filtering: Actual road experiments produce highly noisy data. The noise level varies from road to road and from time to time. Since detection of honk involved identifying some frequencies whose amplitudes are above average amplitudes by a certain threshold, that threshold becomes difficult to be ascertained if the average amplitudes varied. This is exactly the problem posed by variable noise which makes average amplitude to fluctuate. Hence band-pass filtering, that reduces amplitude of frequencies outside the honk frequency range of 2-4 Khz is done in *Audacity* prior to further processing.

Breaking time into small windows: Unlike [7], in our case, the problem is not only to detect the presence or the absence of a honk. In case we detect the presence, we also need to detect the start and end times of the honk as precisely as possible. The first reason for this is matching two honks recorded in two recorders as the same will be difficult if boundaries are imprecise. Secondly, if we do not get the entire honk frequency spectrum, $f1$ and $f2$, extracted from the spectrum, might not be correct. Hence we decide to divide time into small windows to get precise time values.

The algorithm choices for honk detection that we discuss next, involve frequencies in the recording and hence we need to compute FFT. The number of samples in a time window, to be used in FFT computation of that time window, is required to be equal to the number of FFT points. 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. This will decrease number of samples and hence decrease usable number of FFT points. Reducing number of FFT points reduces frequency resolution. Thus, though from the viewpoint of improving time granularity of detected honk, decreasing time window size seems necessary, the effect of decrease in frequency resolution in detection needs to be analyzed. We do this after stating the algorithm choices of honk detection.

6.1 Algorithm choices

We considered three possible choices for the algorithm.

(1) PeakVsAvgAllFreq: This algorithm, similar to [7], 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 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). The intuition behind this is the same as the intuition being bandpassing. We wish to remove effect of noise in the spectrum outside honk spectrum on deciding honk detection thresholds.

(3) PeakAbsAmp: This labels a time window as a honk if the absolute threshold of any frequency in 2-4KHz range exceeds -20dB. We saw the amplitudes of frequencies in honk spectrum from the empirical data and saw that a honk typically has amplitudes $>$ -20dB in the honk spectrum.

6.2 Choosing the time window size

Returning to the question of what time window size to use, we observe the following. In the honk detection algorithms above, the exact value of frequency is unimportant. Whether a peak exists with the property stated in the heuristic is all that matters. Thus high frequency resolution and hence high number of FFT points are not required. We choose to use 128 FFT points (128 is the minimum FFT points supported by the open source FFT implementation we use). With 16 Khz sampling frequency and 128 samples per time window, we have time window size as 8 millisecs which is enough for time granularity.

6.3 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. 5, where we manually (visually and through hearing) labeled 257 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.

6.3.1 Honk length bounding

On closer look, we found that most of the false positives were due to stray windows i.e. some stand alone windows accidentally matching the criteria of the detection algorithm. Since our CDF in Fig. 7 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. False positives are largely reduced.

6.3.2 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. This honk merging step was implemented by Prashima Sharma.

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.

6.3.3 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 T. In fact, these values are for T=2, instead of T=10 as in *PeakVsAvgAllFreq*. Still the false negatives are so high. The false positives for *PeakVsAvgHonkFreq* are low, but having high false negative is a more serious offense in our scenario. Having same false positives in both recorders is very unlikely, so they will get filtered out in the honk matching step. But a false negative cannot be rectified in any way.

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	9.8	43	18.9	0.3
length bounding	5.6	0.7	2.1	74.7	10	1.04
honk merging	5.7	0.4	2.1	73.8	10.3	1.01

Table 1. Comparison of honk detection algorithms

6.4 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.*

6.5 An alternate algorithm choice

Prof. Preeti Rao of speech processing group in the Electical Engineering Department of IIT Bombay suggested an alternate algorithm of honk detection as follows.

Equidistant_maximas Frequencies in 2-4 Khz having highest amplitudes form an arithmetic progression series. See Fig.8, where the arrows between two successive maximas are approximately of equal length.

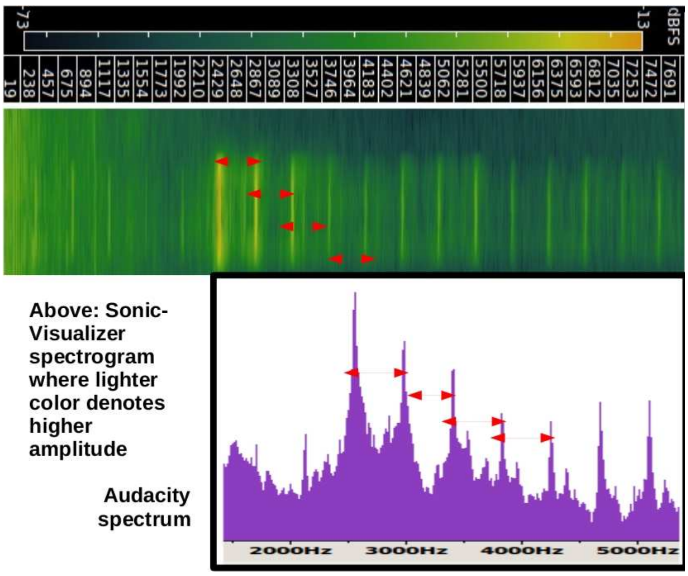

Figure 8. Equidistant peaks

But when frequency spectrum of different honks were seen in *Audacity* and *Sonic-Visualizer*, the number of maximas varied from honk to honk, the distances between consecutive maximas pairs were approximately equal instead of being exactly same and this distance also varied among honks. These factors made it difficult to decide uniform thresholds. Furthermore, high frequency resolution was needed that came into conflict with the need of high time resolution in detection. Hence the algorithm was not used.

7 Honk Matching

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).

To match honks, we consider the following two intuitions.

(1) 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.

(2) DurnRatio: This criterion bounds the ratio of honk durations in the two recorders to be above R. Ideally, if d_1 and d_2 are the honk durations at recorders R1 (honking vehicle receding this recorder) and R2 (honking vehicle approaching this recorder) 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). So, $\frac{d_2}{d_1} = \frac{f_1}{f_2} = \frac{v - v_s}{v + v_s}$ where v is speed of sound and v_s is speed of vehicle. Since v is fixed, this ratio will decrease with increasing v_s. The maximum value of v_s on Indian roads is about 50Kmph. Thus $\frac{d_2}{d_1} > 0.92$ i.e. R = 0.92.

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.

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. 9. 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 experiment was conducted and data analysis was done along with Prashima Sharma.

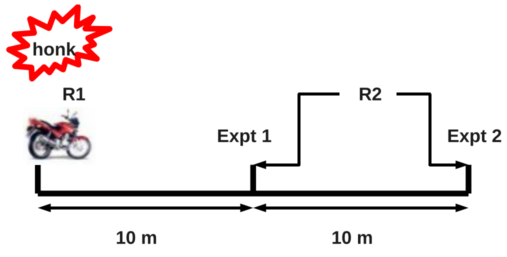

Figure 9. Evaluation setup for *StartTimeDiff* & *DurnRatio*

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. 10 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.

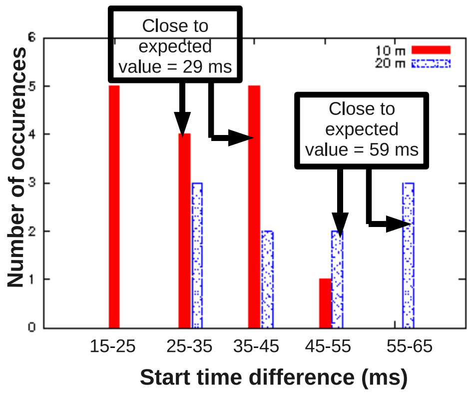

Figure 10. Start time difference values (ms) for 25 honks

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.

We viewed each honk pair in $Audacity$ and found a significant trailing pattern after each honk in Recorder-2 (see Fig. 11). 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

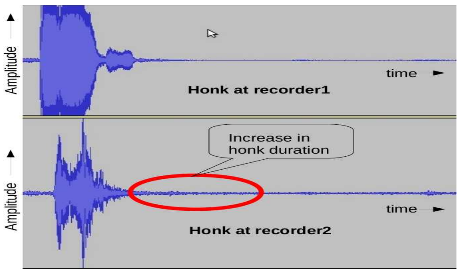

Figure 11. Trailing honk pattern in Recorder-2

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.*

8 Frequency Extraction

From a pair of honks, matched across the two recorders, we need to find $f1$ and $f2$ and calculate speed. With 16 Khz sampling frequency, we have frequencies varying from 0-8Khz in each of the two matched honks after FFT computation. We are thus left with the question of how to choose a single $f1$ and a single $f2$ from this range. In this section, we concentrate on this problem of frequency extraction.

8.1 Choosing FFT Points

In Section 6, we reduce the FFT points to 128 for (a) increasing time resolution at the expense of (b) decreasing frequency resolution. In the detection algorithm, (a) is necessary for accurate honk boundary detection and (b) does not cause problem as precise frequency values are irrelevant.

Can 128 points be used here? - N point FFT computation clubs

$$n = (samplingfrequency)/N \tag{4}$$

frequency values into one bin and the central frequency of that bin represents all the n frequency values. So in Fig.12, frequencies 0 to n will be represented in FFT output as $n/2$ and n to $2n$ as $3n/2$. So if we have $f1 = n/2$ and $f2 = 3n/2$, we calculate the speed value as

$$v_s = \frac{(f1 - f2)}{(f1 + f2)} v \tag{5}$$

while the actual values of the frequencies might be (a) $F1 = 0$ and $F2 = 2n$, in which case correct value of speed should be

$$\begin{aligned} \frac{(F1 - F2)}{(F1 + F2)} v &= \frac{(f1 - n/2) - (f2 + n/2)}{(f1 - n/2) + (f2 + n/2)} v \\ &= \frac{(f1 - f2) - n}{f1 + f2} v \end{aligned} \tag{6}$$

or (b) $F1 = F2 = n$, in which case correct vlue of speed should be

$$\begin{aligned} \frac{(F1 - F2)}{(F1 + F2)} v &= \frac{(f1 + n/2) - (f2 - n/2)}{(f1 + n/2) + (f2 - n/2)} v \\ &= \frac{(f1 - f2) + n}{f1 + f2} v \end{aligned} \tag{7}$$

Since $(f1 - f2)$ is negative, in the former case, we are under estimating speed and in the latter case, over estimating it. The lower the value of N, higher is the value of n and more is the error in speed calculation. So here we should use high value of N. The corresponding decrease in time resolution is not an issue now, as we have already detected the honk windows precisely. So we can compute a high point FFT on the honk windows to extract $f1$ and $f2$ as precisely as possible.

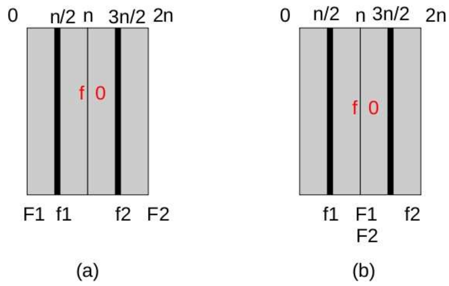

Figure 12. Effect of FFT points on speed

So what N to use? - In choosing high value of N, we need to fulfill 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 256 ms as our sampling frequency is 16 KHz. From Fig.7, about 65% of the honks in each sound clip is less than 250 ms in length, so we will have very few honks with 256 ms time window. Hence we choose N = 2048, which needs 128 ms time window. If a honk has more than one 128 ms windows, then we do 2048-point FFT for each individual window and average out the amplitudes of each frequency across the multiple windows. According to Section 6, the minimum honk duration for us is 112 ms. 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 will use N = 1024.

8.2 Algorithms

1) peak_remains_peak - Spectrum of several pairs of matched honks in $Audacity$ show local maximas remain similar across matches (see Fig. 13). Based on this observation, we argue that if a certain frequency has highest amplitude in a honk and another frequency has the highest amplitude in its matched counterpart, these two frequencies are the Doppler shifted version of the same frequency in the original honk. This is intuitive theoretically as well. Doppler shift changes the value of a frequency based on speed of source or receiver of sound, it does not affect the amplitude of that frequency in any way. So we choose frequency having highest amplitude from one honk as $f1$ and frequency having highest amplitude from its matched counterpart as $f2$. Using these, we compute speed according to Equation 3.

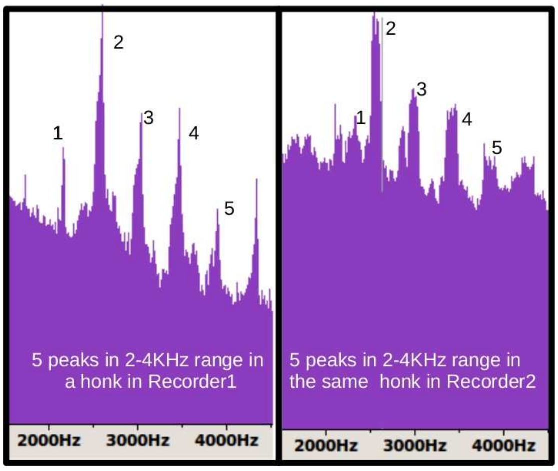

Figure 13. Local maximas remaining unchanged after Doppler shift

2) peak1_peak2_exchange - After analyzing data from the experiment **campus_speed_vary** described in Section 9.2.1, Prashima Sharma observed that in 3 cases out of 25, the second highest peak in one honk becomes highest peak in its corresponding match. We call this situation **peak1_peak2_exchange**, which is described in Fig. 14. We argue that since

$$f1/f2 = (v - v_s)/(v + v_s) \tag{8}$$

where v is speed of sound and v_s is speed of the honk source, we can find a lower bound of $f1/f2$ by assuming an upper bound for v_s. The latter on most Indian roads is about 50Kmph, though it can be varied from historical knowledge of speed values of any road. For example, with $v = 340$ m/sec and upper bound of v_s as 50 Kmph, $f1/f2$ has lower bound of 0.92.

So 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. 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* **peak1_peak2_exchange** *heuristic, and use Eqn. 3 for speed estimation.*

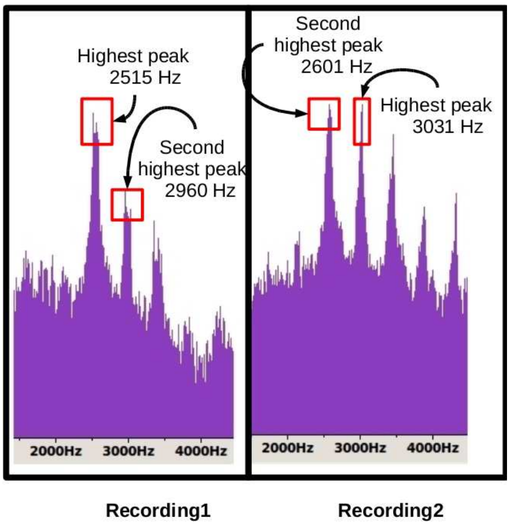

Figure 14. Exchange of first and second maximas after Doppler shift

9 Experimental evaluation of speed estimation technique

Next we seek to experimentally evaluate our three step speed estimation technique. We do semi-controlled experiments in campus and on real roads, both *Adi* and *Hira*, using our own bike and honks from it.

9.1 Ground Truth

9.1.1 Detection and matching

In evaluating honk detection and honk matching, simple manual annotation of the recording suffices. For example, when experiments are done on real roads, there are many vehicles honking, other than our own bike. We wish to evaluate honk detection and matching based on the honks of our own bike. So every time our bike honks, one of the two persons holding N79s, speaks into the phone giving details of that honk. Later these manual annotations are used to filter out the extra honks. Fig. 15 shows how a honk followed by manual annotation looks in *Praat*.

9.1.2 Speed ground truth issues

But speed ground truth is more difficult to ascertain. Speedometer errors, parallax error while reading make it difficult for even drivers of our bike to tell the speed with confidence. It is risky to do the three tasks of maintaining a constant speed, blowing a honk and reading the speedometer simultaneously while driving on a real road with other vehicles around.

In our setup, 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. Thus now we can have three estimates of speed for each honk: 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,

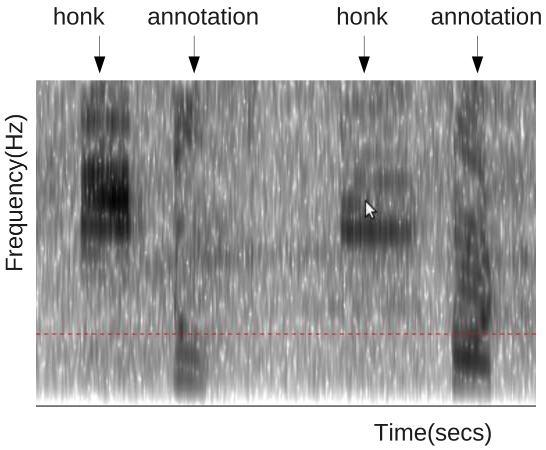

Figure 15. Honk and manual annotations in Praat

which we term v_{13}. We get v_{23} similarly from Recorder-2 and Recorder-3, using Eqn. 1.

Intuitively, less the difference between the three speed estimates from the same honk, more is the reliability of our technique. However it might happen that though the three estimates are close to each other, all three are in large error in terms of the actual ground truth. Hence we do a sanity check by comparing the mean of the three speeds to the speed intuitively estimated by the driver.

9.2 Experimental setup

9.2.1 Campus-road experiments (campus_speed_vary)

On a campus road, our bike was driven past the sensors at various speeds. We varied the speed from 0 Kmph (stationary), to slow (about 10 Kmph), to medium (about 25 Kmph) to high (about 35 Kmph). The vehicle blows a honk near the middle of the two recorders. A total of 30 honks are blown in 30 different experimental runs. The experiments were done on 24^{th} Sept and 14^{th} Oct, 2009 late in the night, to ensure absence of other vehicles.

9.2.2 City-road experiments (road_speed_vary)

We conducted similar experiments at roads *Hira* and *Adi* too. Here too, we varied the motorbike speed between 0 Kmph and about 40 Kmph (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. These experiments were done on 28^{th} Nov, 2009.

9.2.3 Varying the position of vehicle honk (road_position_vary)

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. 16. 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. These experiments were done on 28^{th} Nov, 2009.

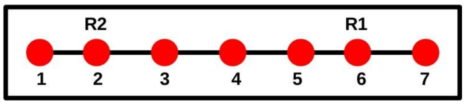

Figure 16. Honking positions of bike

9.3 Results

9.3.1 Detection and matching in the campus_speed_vary and road_speed_vary experiments

In the **campus_speed_vary** experiment, 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. In **road_speed_vary** experiment, 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.

9.3.2 Speed estimates in the campus_speed_vary and road_speed_vary experiments

Table 2 show the speed estimates for the 25 honks matched across all three recorders in the **campus_speed_vary** experiment. The mean speeds conform to the intuitive speed estimates given by the driver in most cases. The maximum standard deviation across v_{12}, v_{13} and v_{23} is 5.44 Kmph which shows that the three estimates are fairly close. Table 3 show the speed estimates for the 26 honks matched across all three recorders in **road_speed_vary** experiment. The mean speeds conform to the intuitive speed estimates given by the driver in most cases. The maximum standard deviation across v_{12}, v_{13} and v_{23} is 3.69 Kmph which shows that the three estimates are fairly close.

9.3.3 Speed error metrics for the campus_speed_vary and road_speed_vary experiments

Apart from standard deviation, we use three different measures of error in the speed estimates. We define **Avg3Err** as the average of the three error quantities $|v_{12} - v_{13}|$, $|v_{13} - v_{23}|$, and $|v_{23} - v_{12}|$. And **Max3Err** as the maximum of these three quantities. Apart from these two measures of error, we estimate the **relative error** by taking the ground truth of the speed to be the average of v_{13} and v_{23}.

For each of the 25 matched honks in the **campus_speed_vary** experiment, Fig. 17 shows the three measures of error. *Avg3Err* and *Max3Err* are given on the left y-axis, while the relative error is given on the right y-axis.

type	v12	v13	v23	mean	s.d.
sta	0	0	0	0	0
sta	0	0	0	0	0
sta	0	0	0	0	0
sta	1.89	7.57	3.77	4.41	2.89
slow	11.2	14.79	7.53	11.17	3.63
slow	13	18.38	7.5	12.96	5.44
slow	14.88	18.38	11.3	14.85	3.54
slow	14.93	18.43	11.33	14.9	3.55
slow	16.82	18.43	15.16	16.8	1.64
med	20.55	19.03	22.03	20.53	1.5
med	20.56	22.05	19.01	20.54	1.52
med	20.56	19.01	18.93	19.5	0.92
med	22.39	22.81	21.99	22.4	0.41
med	22.39	22.81	21.99	22.4	0.41
med	23	15.62	25.19	21.27	5.01
med	23	25.08	26.27	24.78	1.65
med	24.22	22.81	25.58	24.2	1.38
med	25.5	26.04	24.98	25.51	0.53
med	25.5	20.83	29.98	25.44	4.57
high	27.99	20.83	20.23	23.02	4.32
high	27.99	20.83	34.83	27.88	7
high	30.73	29.98	31.52	30.74	0.77
high	33.36	34.97	31.66	33.33	1.66
high	33.36	31.66	34.97	33.33	1.66
high	33.36	29.98	36.93	33.42	3.48

Table 2. Speed Estimates in Kmph the campus_speed_vary experiment

type	v12	v13	v23	mean	s.d
sta	0.00	0.00	0.00	0	0
sta	0.00	0.00	0.00	0	0
sta	0.00	0.00	0.00	0	0
sta	0.00	0.00	0.00	0	0
sta	0.00	0.00	0.00	0	0
sta	0.00	0.00	0.00	0	0
sta	4.65	3.11	7.05	4.94	1.99
slow	9.50	8.75	11.46	9.9	1.4
slow	13.95	10.58	17.96	14.16	3.69
slow	13.95	10.58	17.24	13.92	3.33
slow	14.11	17.43	15.15	15.56	1.7
slow	14.11	17.44	10.71	14.08	3.37
slow	14.15	13.99	14.32	14.15	0.16
slow	14.15	13.99	14.32	14.15	0.16
slow	14.15	13.99	14.32	14.15	0.16
slow	15.30	12.39	17.29	14.99	2.46
med	17.49	17.24	17.74	17.49	0.25
med	19.10	17.63	17.95	18.23	0.77
med	21.04	18.93	17.84	19.27	1.63
med	22.00	24.27	21.60	22.62	1.44
med	22.00	24.20	21.54	22.58	1.43
med	22.96	20.75	21.80	21.84	1.11
med	24.60	24.14	21.47	23.4	1.69
high	26.42	24.14	28.80	26.45	2.33
high	26.42	27.51	28.80	27.57	1.19
high	29.00	34.19	28.80	30.66	3.06

Table 3. Speed Estimates in Kmph for the road_speed_vary experiment

The points on the x-axis are sorted in increasing order of relative error.

Similar error plots are shown for the 26 matched honks in the **road_speed_vary** experiment in Fig. 18.

We see that both in terms of absolute error and the relative error, our mechanism is quite reliable, even in noisy city road conditions. The *Avg3Err* and *Max3Err* measures are mostly

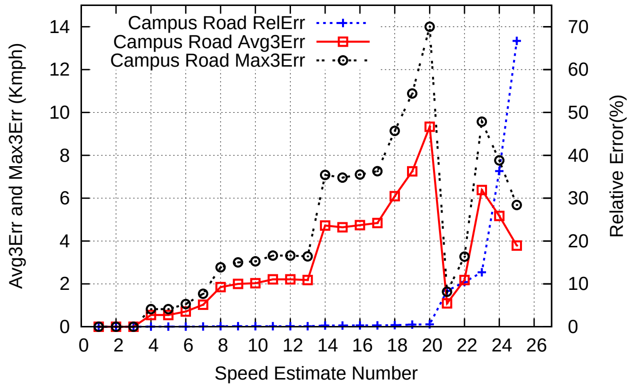

Figure 17. Speed estimate errors on campus road

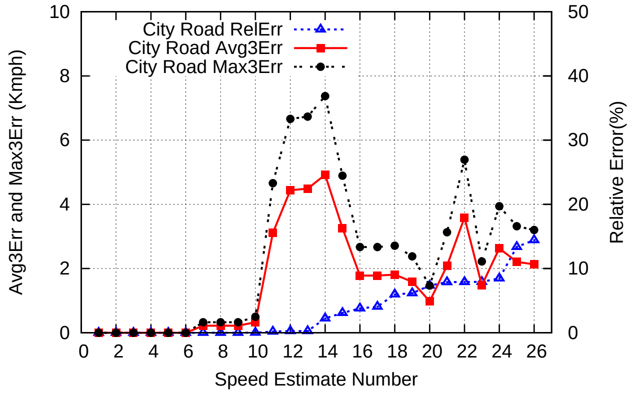

Figure 18. Speed estimate errors on city road

under 5-10Kmph. The relative error is mostly under 10%. There is one case of high relative error of about 65%. As can be seen, the *Avg3Err* and *Max3Err* are low for these cases. We also verified that this was a case where the absolute speed itself was low, and hence the relative error is high.

9.3.4 Speed error metrics for the road_position_vary experiment

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. 19 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 Kmph and the *Max3Err* is below 10 Kmph 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

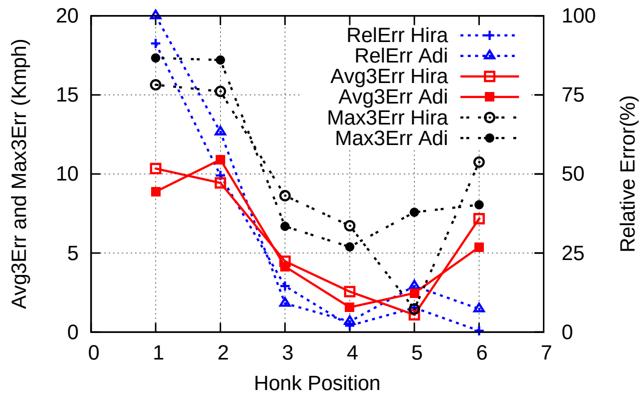

Figure 19. Speed estimate errors at various honk positions

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.

9.4 Discussion

- Detection and frequency extraction steps in the three step speed estimation process are seen to be working well according to the above results. The matching algorithm is somewhat naive, just comparing the start time differences to a threshold. Even deciding this threshold is difficult due to the variability of propagation delay of sound. We plan to study speech processing techniques that match the same sound based on frequency patterns and see their applicability in honk matching.

- Some better mechanism to ascertain speed ground truth needs to be developed. Only then we can quantify the true accuracy of our speed estimates.

In the next section, we shall see how we can work around the wrong speed values coming from bad matches, and estimate traffic state despite some fraction of errors in vehicular speed samples.

10 Traffic state classification

Now if we put the roadside recorder pairs on real roads and record honks from actual road vehicles, will the speed estimates obtained from these recordings portray the contemporary traffic state? If that is possible, then useful mobile applications can be developed to give traffic updates and travel time estimates 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 based on speed estimates. We eventually develop some non speed based acoustic metrics that can be used in conjunction with speed based metrics in traffic state classification.

10.1 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, (see Figure 20), versus free-flowing (see Figure 21).

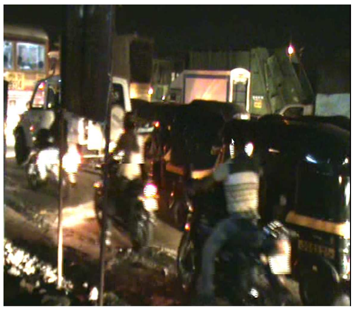

Figure 20. Congested traffic in *Adi*

Figure 21. Free-flowing traffic in *Adi*

The details of dates and times of experiments in *Hira* are given in Table 4. 'u' denotes uncongested or freeflowing and 'c' denotes congested traffic. Hira1 refers to one road in *Hira* that always has freeflowing traffic in both directions. Hira2 refers to a road in *Hira* that has congested traffic in one direction and free flowing in the other direction sometimes. In case of Hira2, traffic state is thus represented as combination of two states, one in each direction. At *Adi*, we collected 4.5

Date	Time	Venue	Traffic State
2^{nd} Nov	8 – 9 pm	Hira1	u
4^{th} Nov	7.30-8.30 pm	Hira1	u
6^{th} Nov	6.30–7.30 pm	Hira2	c+u
7^{th} Nov	5 – 6 pm	Hira1	u
7^{th} Nov	6.20-7.20 pm	Hira2	c+u
9^{th} Nov	5 – 6 pm	Hira1	u
9^{th} Nov	6.20-7.20 pm	Hira2	c+u
11^{th} Nov	6.20-7.20 pm	Hira2	c+c
12^{th} Nov	9 – 10 pm	Hira2	u

Table 4. Experiment date and time at Hira

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. The details of dates and times of experiments with corresponding traffic states are given in Table 5. As mentioned earlier, both

Date	Time	Traffic State
16th Nov	3.25-4.25 pm	u
16th Nov	7.15-8.15 pm	c
17th Nov	3.15-4.15 pm	u
17th Nov	7.15-8.15 pm	c
18th Nov	4.30-5.30 pm	u
18th Nov	5.30-6.30 pm	u
18th Nov	6.30-7.30 pm	c
4th Dec	6 – 7 pm	30 mins u + 30 mins c
4th Dec	7 – 8 pm	c

Table 5. Experiment date and time at Adi

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*.

10.2 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.

10.2.1 Granularity of classification

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.

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. 22 and Fig. 23 show 6 sample CDF plots each ($10min \times 6 = 1hr$ each), under congestion and free-flowing traffic, on *Adi*. The congested plots are of 16^{th} Nov, 2009, 7.15-8.15 pm, and free-flowing are of 18^{th} Nov, 2009, 5.30-6.30 pm.

Thus when we do traffic classification later, we do it in blocks of 10 mins. We take a 10 min block and classify it as congested or free-flowing based on some metric values.

10.2.2 Observations from CDF plots

From the various CDFs of 10-min durations (only 12 of which are shown in Fig. 22 & 23), we observe the following.

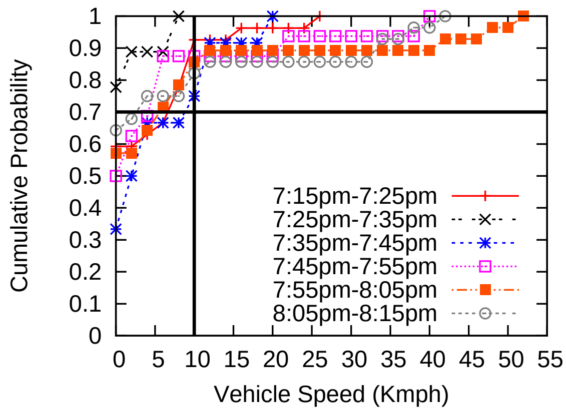

Figure 22. Speed CDF samples: congested traffic in *Adi*

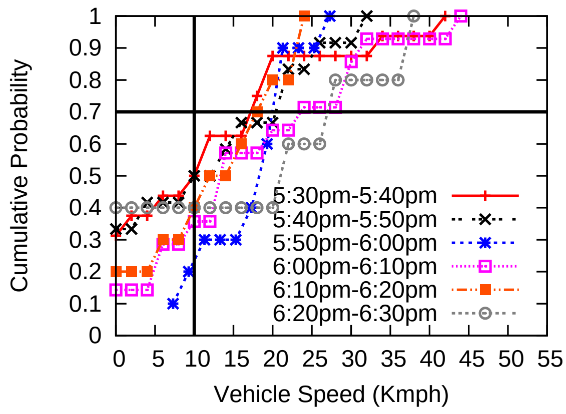

Figure 23. Speed CDF samples: free-flowing traffic in *Adi*

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 different 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 [6]). (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 in our honk-based speed sampling, towards lower speeds. Another reason is that, like in Fig. 16, 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. 10.3) shows how we can work around this.

10.2.3 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. 24 and Fig. 25 respectively which belong to 9^{th} Nov, 2009, 6.20-7.20 pm. 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, 2009. 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. 26 and Fig. 27.

10.3 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. 22 & 23. We present two kinds of metrics: (a) speed-based and (b) non-speed based acoustic metrics.

10.3.1 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) $P(v_s < 10Kmph)$, that is percentile of speed samples less than 10 Kmph. 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. 22 versus Fig. 23. The 70th percentile horizontal line and the 10Kmph 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. 6.

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.

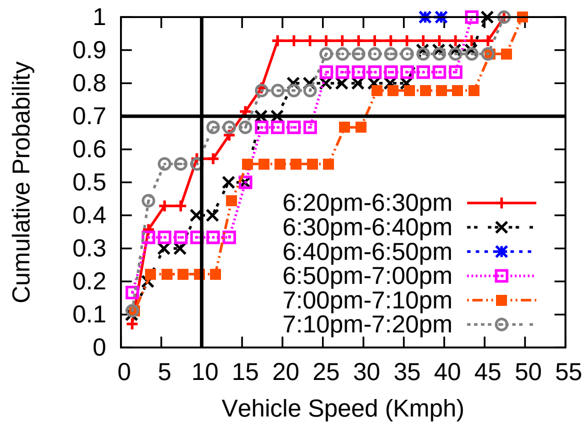

Figure 24. North-South Direction on Normal Day

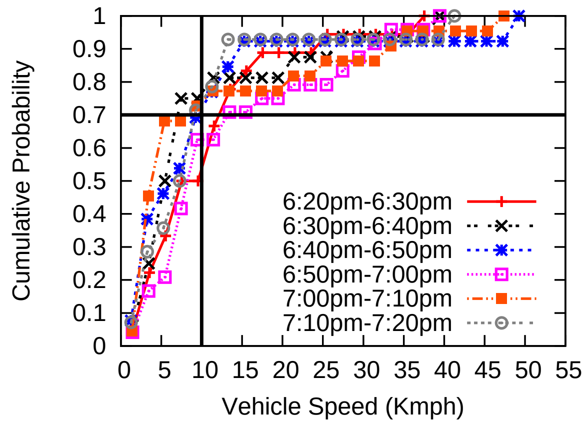

Figure 25. South-North Direction on Normal Day

	Hira		*Adi*	
Metric	Congested mean (s.d) [30 samples]	Free-flow mean (s.d) [48 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 6. Speed based metrics

10.3.2 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.

Tab. 7 shows the mean across the various 10-min samples as well as the standard deviation, of the three metrics for

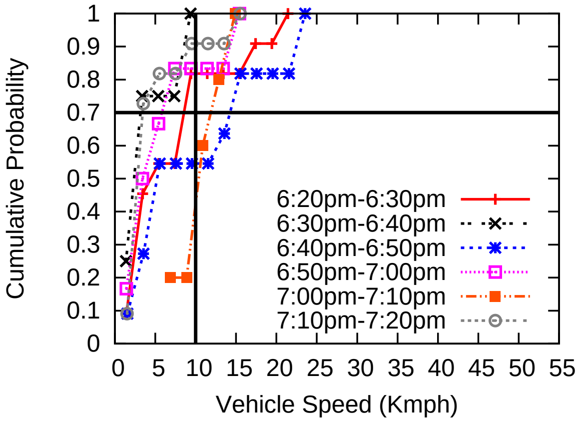

Figure 26. North-South Direction on Rainy Day

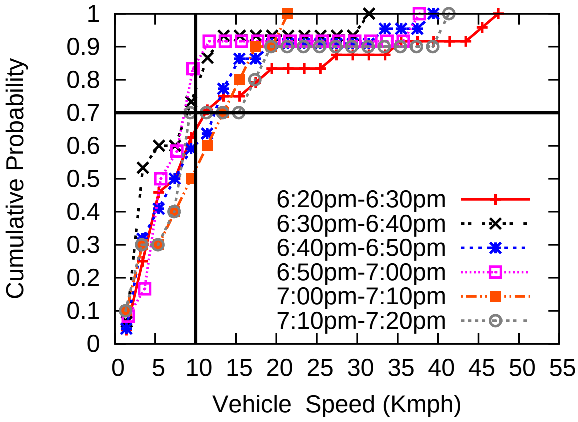

Figure 27. South-North Direction on Rainy Day

	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 7. Non speed based acoustic metrics

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*.

10.3.3 Discussions on metric choice for binary traffic classification

After seeing the five metrics, two speed based and three non speed based, we conclude the following.

- The non speed based metrics are direction insensitive. So we cannot use them for the directionwise traffic state classification.

- Two or more the metrics should be used in conjunction with one another to decide the traffic state. Any one can give false alarms. For example, during free flow, it might happen that, none of the drivers of fast moving vehicles honk. Only one or more vehicles, that slow down or stop for some reason, blow honk. Our speed based metrics will give very pessimistic view of the traffic state in this case. But if the non speed based metrics are also considered, the actual scenario will become apparent.

- Metric should be chosen specific to road stretch, noise level can be a metric in *Hira*, but not in *Adi*.

- The metric mean values, as seen from the tables, indicate that thresholds for each metric should be road specific. If we always use number of honks under congestion in *Adi* as threshold, *Hira* will always seem to have free flow, as traffic load is much less there, even under congestion.

10.4 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 30 samples from the congested state and 48 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. 8 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.

10.5 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. 6 & 7 for the 5 metrics.

	Mann-Whitney U test		Kolmogorov-Smirnov test	
Metric	*Hira*	*Adi*	*Hira*	*Adi*
70th 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 8. p-values of statistical tests

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, $X_{thr} = (7.7 + 21.1)/2 = 14.4Kmph$ 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 Kmph, it would be classified as free-flowing, and as congested otherwise.

The various metric mean values, as seen from Tab. 6 & 7, 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 9. 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. 9 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%. Noise level metric in *Adi* gives high error as expected.

10.6 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 Kmph are plotted in Fig. 28. There are 12 values for each metric, corresponding to 12 clips of 10 mins, over 2 hours.

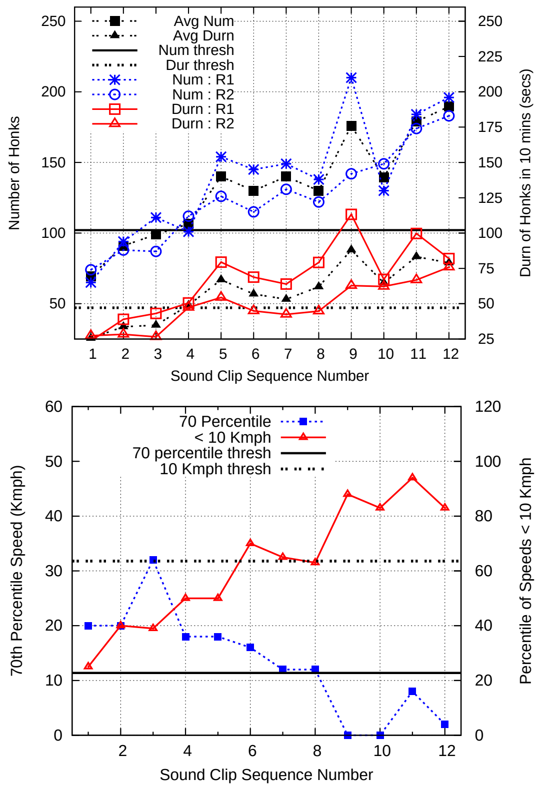

Figure 28. Change in metric values in two hours

The figures also show as horizontal lines, the classification thresholds computed, as per Sec. 10.5. For this, we use all other data on *Adi*, except these two hours, as training set.

10.6.1 Observations

The main observations from the figures are as follows.

- The plots in Fig. 28 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*, $P(v_s < 10Kmph)$, *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. Another possible reason for the number of honks detected at R1 being higher is the presence of an impending fork in the road just ahead of R1. Vehicles honk more as they approach the fork.

- We make a final observation using the above data. 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 Kmph, and 80% of the speeds are < 10 Kmph. Thus the clip clearly belongs to congested state. This supports our earlier conjecture that metrics used in conjunction with one another provide more information than using them individually.

10.6.2 Inferences

Observing the onset of congestion manually, we could intuitively understand the reason for the rapid change in the traffic state and even think of some simple and obvious solutions to it. The road had an impending signal about 300 m ahead of our recorders' position. At a signal, red light duration is the time when the traffic queue builds up, and green light duration is the time, when the traffic queue is cleared. Initially, in the non peak hours, the small number of vehicles that join the queue during a red light, get enough time to leave the queue during the next green light. Gradually fill rate starts growing while service time remains same and the queue cannot be cleared within that service time. The vehicles remaining to be cleared in each green cycle add to the queue length to be cleared in the next green cycle and the situation worsens rapidly with a long queue building up. Gradually when fill rate again comes down, the vehicles get cleared. Given that the peak hours when people go to work in the morning or return from work in the evening span 2-3 hours, conditions at important traffic signal points is bound to be grave, which is what we regularly observe in reality.

In case of this particular road stretch, *Adi*, the road perpendicular to it, that shares the same signal and operates in the alternate cycles, is a low vehicle load road and the load remains mostly constant throughout the day. It does not connect any important points that will cause traffic to grow during peak hours. On the other hand, traffic on *Adi*, as we have noted, increases drastically in the peak hours. So a simple arrangement to adjust the signal timer, to have a longer green cycle in *Adi* during peak hours, than on its perpendicular road stretch can improve the situation. Another purpose the signal serves is to allow people to cross *Adi*. A foot bridge or a subway is all that is needed to remove this second factor.

11 Some Problems Faced

Certain issues slow down the course of our work at times. We discuss two such problems here.

11.1 Why Does Recording Stop?

On certain occasions, the recordings in the two N79s, even after synchronization using the method discussed in Section 4.5.1, became unsynchronized after recording for a while. We first detected this problem when the same honk, recorded in the two phones held close to each other, showed a start time difference as high as about 2 secs in the two recordings. Multipath reflection or variability of sound propagation speed would not cause such high delay. The problem seemed to be that any one N79, in course of recording, stopped recording for say Δt time and then resumed to record. This resumed point of recording in this N79 was Δt time behind its corresponding point in the other N79 recording. The behavior was sporadic and hence difficult to reproduce and analyze. With the intuition that some other process is interfering with the recording, we turned off processes like screen reorientation using accelerometer and Wi-Fi access point probing, searched for unnecessary background processes running on the phones using a software $Taskman$ and killed them. We tried to see if moving the phones causes the problem. Finally we realized that lighting up the display by pressing the scroll key stopped recording. 40 sets of experiments were done to validate that the problem occurred if key is pressed and another 40 sets to validate that problem did not occur if key is not pressed. These experiments and analysis were done along with Prashima Sharma.

11.2 Are We Terrorists?

All our road experiments were done in Mumbai, an Indian city that has undergone a number of terrorist attacks in the recent past. So sitting with mobile phones on the roads, doing recording for hours, naturally raised suspicion of pedestrians who informed security officers that we were doing something fishy. Explaining our goal to such individuals and convincing them by showing our credentials required time and patience.

12 Conclusions

The important conclusions of our work are as follows.

- ITS techniques are needed in developing regions to alleviate traffic issues. Existing ITS techniques are difficult to deploy in developing regions for several reasons. Hence designing and implementing a new ITS technique is an important problem to solve. The chaotic nature of traffic and low cost constraints in developing regions, make the problem challenging and hence interesting.

- We can estimate speed of vehicles from honks using a two sensor architecture and suitable algorithms for honk detection, honk matching and frequency extraction. The matching algorithm is somewhat naive and gives some false matches. Still the estimated speeds have error under 6 Kmph, if we take the standard deviation among three estimates of the same speed as error.

- The empirical CDF of speed values estimated from city road recordings over 18 hours duration, in blocks of 10 minutes, brings out the traffic state of those 18 hours as congested or freeflowing.

- Though there are some outliers in the speed CDF's, metrics like a) 70^{th} percentile speed and b) percentile of speed < 10 Kmph can be used to do binary classification of traffic states into congested and freeflowing. Some non speed based acoustic metrics like c) number of honks d) duration of honks and e) noise level can also be used for such classification. Metrics should be used in conjunction with one another to increase classification accuracy.

- Directionwise traffic state identification is possible on a bidirectional road using the speed based metrics as our speed estimates are signed, the sign giving the direction of vehicle motion.

- The values of all the metrics for congested and freeflowing states are statistically different. This has been verified at 99% confidence level using Mann-Whitney U test and two sample Kolmogorov-Smirnov tests.

- A threshold based classification of traffic state into congested and freeflowing has been trained and tested with n-fold cross validation using 18 hours of city road data where minimum classification accuracy was about 75%.

- Onset of congestion can be detected based on rise or fall of metric value above or below the threshold for that metric.

- Our system will be low cost - with each sensor costing around $20.

13 Future Work

There are several things that are needed to be done. We divide them into short and long term goals. The short term goals should be met in the next 4-5 months time. The long term goals should be studied and understood in parallel to implementing the short term goals and gradually worked on in the next 2-3 years.

Short Term Goals:

- Implementing a better matching algorithm or at least tuning the matching parameters more carefully to filter out spurious matches.

- Implementing a technique to know vehicle speed ground truth accurately.

- Designing low cost acoustic sensors customized for our purpose.

- Implementing connectivity and data transfer from the recorders in the experimental site to central servers for real time analysis.

Long Term Goals:

- Optimal sensor placement based on traffic criticality or road coverage.

- Automating the threshold calculation for different metrics on new roads using some clustering mechanism.

- Correlating data from different sensor pairs and designing algorithms to estimate travel time.

- Correlating data from consecutive sensor pairs to estimate vehicle queue length

- Time series analysis to detect congestion patterns at specific times of the day.

- Forecasting future traffic state from present state using machine learning techniques on historical data.

- Designing user level applications to provide information to commuters on the road.

Acknowledgment

This work is supported in part by Microsoft Research India. I thank my advisor Prof. Bhaskar Raman for not only providing ideas and guidance but also for working with me all through. I thank Prashima Sharma, my project teammate for sharing a large part of the work. I thank Prasad Gokhale, Prof. Om Damani, Amit Srivastava, Ajinkya Joshi, Piyali Dey, Akash Sharma, Vijay Gabale and Lokendra Kumar Singh, without whom the experiments would not have been possible. I thank Chitralekha Gupta for her earnest help regarding audio processing softwares. Zahir Koradia and Prof. Preeti Rao provided valuable input at the inception of this project.

References

[1] http://www.nhai.org/roadnetwork.htm.

[2] http://www.neoncarrot.co.uk/h_aboutindia/india_statistics_1.html.

[3] http://en.wikipedia.org/wiki/Infrastructure_in_Bangalore.

[4] http://en.wikipedia.org/wiki/Interstate_Highway.

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

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

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

[8] 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.

[9] 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.

[10] Jungkeun Yoon, Brian Noble, and Mingyan Liu. Surface street traffic estimation. In *MobiSys '07*. ACM, 2007.

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

[13] 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.

[14] `http://traffic.berkeley.edu/theproject.html`.

[15] Y. Ramakrishna, P. Ramakrishna, V. Lakshmanan, and R. Sivanandan. Use of gps probe data and passenger data for prediction of bus transit travel times. In *Proceedings of 2007 Transportation Land Use, Planning and Air Quality Conference*, July 2007.

[16] Katrina LaCurts Samuel Madden Hari Balakrishnan Sivan Toledo Arvind Thiagarajan, Lenin Ravindranath and Jakob Eriksson. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems(Sensys)*, November 2009.

[17] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *SenSys '08:*. ACM, 2008.

[18] `http://www.merucabs.com/corporate_information.asp/`.

[19] `http://www.easycabs.com//`.

[20] `http://www.bmtcinfo.com/site/index.jsp`.

[21] `http://en.wikipedia.org/wiki/List_of_most_populous_cities_in_India`.

[22] `http://www.financialexpress.com/news/by-next-year-well-target-10-of-indias-smartphone-market/551198/0/`.

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

[24] 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.

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