Soft-Swipe: Enabling High-Accuracy Pairing of Vehicles to Lanes using COTS Technology

Gopi Krishna Tummala
The Ohio State University
tummala.10@osu.edu

Derrick Cobb
Honda Engineering North America, Inc.
Derrick_Cobb@ega.honda.com

Prasun Sinha and Rajiv Ramnath
The Ohio State University
{sinha.43, ramnath.6}@osu.edu

ABSTRACT

Proximity-based interactions underlie applications such as retail payments using smartphone-apps like Apple Pay and Google Wallet, automated grocery store checkout, and vehicular transactions for toll payments. In these applications, a transaction takes place between two objects when they come close to each other, with relative proximity determining the pairing. In this paper, we present a new approach to enable highly accurate pairing of vehicles to specific lanes in a wide-range of vehicle-based multi-lane service stations using general-purpose commodity communication and sensing technology. To evaluate its performance, we consider an example application of pairing vehicles to respective quality check bays in an automobile manufacturing plant. Our proposed system called Soft-Swipe works by matching natural signatures (specifically, motion signatures) generated by the target object with the same signature detected by simple instrumentation of the environment (a video camera or an inexpensive sensor array). Soft-Swipe implemented in a vehicle testing station performed pairing with median F-score of 96% using vision-only system, 92% using sensor-only system and, 99% using both.

1. INTRODUCTION

Smartphone based payments are becoming the new normal as evidenced by ubiquitous nature of mobile payment systems such as Google Wallet and Apple Pay [1, 6]. Banks such as Mastercard and Visa are already working closely with a number of handset developers to make it widely available [9]. These solutions practically work for a few centimeters of range [18] which provides a level of security to the transaction. But, the ability to communicate over longer distances can lead to reduced service time and it can open up opportunities for many new applications.

In this paper we explore applications in which interactions originate from within a vehicle. Transacting from within a vehicle can lead to shorter wait times and higher system throughput. Further, in many situations, the user would be thankful for reduced exposure to inclement weather conditions. The applications can be broadly categorized as follows. **Class-I (Temporary infrastructure):** Parking payments for temporary events such as football games, circus, fair etc. are usually processed manually (both payer and payee) and easily lead to heavy backlog in traffic whose effect can extend for several miles. **Class-II (Small-scale infrastructure):** Application scenarios where the infrastructure is owned by small players can be categorized as follows: *1) Vehicle-specific services:* Payment for services such as car-wash, automated fueling, automated swapping of car batteries for Electric Vehicles (EVs), automated battery charging centers for EVs, and parking charges can be made from within the vehicle. In an automotive manufacturing plant, a vehicle arriving at a manufacturing station needs to be correctly identified so that the appropriate set of tests can be conducted and the appropriate actions can be taken by the assembly line robots or humans. *2) User-specific services*: Payment for drive-thru services such as fast-food, or DVD rental can be supported by such a system. A bank customer can perform automatic verification from inside the vehicle before reaching the ATM machine. Today for such applications usually the payer stops the vehicle to use a machine to make the payment. **Class-III (Large-scale infrastructure):** Highway toll collection systems can afford to deploy various types of expensive equipment such as directional RFID readers, laser sensors and inductive loops. Widely used examples of such systems include E-Z Pass [2], Fastrack [5] and I-PASS [7]. Advanced systems on many US highways do not even require the vehicles to slow down when passing through such checkpoints.

Although for Class III applications a number of solutions are already in place, there are few solutions available for the other two classes. In some cases Class II applications have resorted to using expensive Class III solutions (e.g., JFK airport parking payment lanes offer an option for using E-Z Pass). *This paper presents a first vehicle-to-infrastructure (V2I) pairing system targeting Class I and Class II applications by achieving design goals of low-cost and high-accuracy.* Vehicles that are not paired are to be processed via manual intervention, incidences of which must also be kept to a minimum.

Low cost and limited instrumentation of the infrastructure are the desired criteria for the Class I and Class II applications. The existing solutions for Class III applications such

as E-Z Pass, Fastrack and I-PASS are not readily usable by the other two classes of applications due to the following limitations. *(i) Tag identity database access:* For performing an electronic transaction or authenticating by reading a tag's identity, the system needs access to a database holding the association information with users identity and banking information. In addition, there may be multiple such databases as there are a variety of available toll payment tags [5, 7, 2]. *(ii) Hardware requirement on user end:* The vehicle needs to have a device or sticker placed near the windshield or dashboard. Such placements are prone to mounting errors [3] and the involvement of an additional device at the user end limits its flexibility, since deployment is a custom effort and upgrading the hardware is cumbersome. *(iii) Limited accuracy:* Due to the transmission range of the tags, in scenarios with narrow lanes the signal can be picked up by multiple tollbooths leading to inaccurate charges and unhappy customers [4]. Additionally, the use of such tags for general purpose applications can raise *privacy concerns*[14].

Low cost and limited instrumentation of the infrastructure are the desired criteria for the Class I and Class II applications. The existing solutions for Class III applications such as E-Z Pass, Fastrack and I-PASS are not readily usable by the other two classes of applications due to the following limitations. *(i) Tag identity database access:* For performing an electronic transaction or authenticating by reading a tag's identity, the system needs access to a database holding the association information with users identity and banking information. In addition, there may be multiple such databases as there are a variety of available toll payment tags [5, 7, 2]. *(ii) Hardware requirement on user end:* The vehicle needs to have a device or sticker placed near the windshield or dashboard. Such placements are prone to mounting errors [3] and the involvement of an additional device at the user end limits its flexibility, since deployment is a custom effort and upgrading the hardware is cumbersome. *(iii) Limited accuracy:* Due to the transmission range of the tags, in scenarios with narrow lanes the signal can be picked up by multiple tollbooths leading to inaccurate charges and unhappy customers [4]. Additionally, the use of such tags for general purpose applications can raise *privacy concerns*[14].

Although knowledge of location obtained from the GPS on our smartphone can be used to address the challenges, its accuracy ranges from a few meters to tens of meters [31]. It can perform even poorly near large buildings and concrete structures. Thus, it is not well suited for our needs. Optical Character Recognition (OCR) based number plate recognition systems can be used to detect and identify a particular vehicle. But such a technique requires a dedicated Infra Red (IR) capable expensive camera aiming for a number plate. Additionally, number plate can be occluded by other vehicles in dense class-I and class-II applications.

The necessity of additional hardware can be addressed by developing the smartness as part of a smartphone based application. But the challenge in performing interactions using a longer range WiFi (or similar) technology is the *accurate identification* of the specific device to pair with, from a large number of in-range devices. In particular, financial transactions are *location-aimed* in order to charge the vehicle in a particular lane and position for the provided services. An up-to-date map of all the vehicles can be used to solve the problem. However, the required accuracy calls for techniques that require major hardware upgrades in both the Access Points (APs) and the smartphones, making it difficult to deploy in practice [30, 26, 42]. In this paper we exploit a distinct property of Class I and Class II applications: *slow and time-varying speed of the vehicles*. We refer to the recent time series of velocities of a vehicle as its *motion profile* or *motion signature*. Our solution uses self-generated natural signatures (specifically, motion signatures) reported by the target object matched with the same signature detected by simple instrumentation of the environment (a video camera and/or an inexpensive sensor array), layered on commodity, general-purpose communication and sensing technology (smartphone or other similar device with low-cost inertial sensors) to identify a specific vehicle at a given location (e.g., vehicle A is in lane-4 and next to gate). Our system comprises of two components: (i) A smartphone connected to the vehicle system using a Bluetooth or an OBD-II link or 802.11p link so that it can access the motion profile of the vehicle; (ii) A camera which might be already deployed for security purposes (and/or) (iii) A sensor array deployed for pairing with the vehicles in the lane.

The advantages of our system are many. Unlike range-based pairing technologies such as Near Field Communication (NFC), our system can use any long range radio based communication technologies. Soft-Swipe needs infrastructure areas to be instrumented with commodity products and vehicles equipped with smartphones. Therefore the overall cost of deployment is much lower. Finally, since the device in the vehicle (smartphone) can be programmed, we have the ability to personalize the interactions, such as by allowing the driver to provide additional input, providing status updates to the driver, and so on, as well as to instantly deploy the application and updates.

Soft-Swipe makes the following contributions to the field: (a) presents automatic calibration techniques for infrastructure sensors such as camera and sensors array exploiting Vehicle to Infrastructure (V2I) links. (b) presents sensors fusion techniques by studying individual sensor characteristics. (c) presents a configurable matching system with precision touching 100% for reliable financial transactions and, (d) shows results from extensive evaluation using real world experiments. Note that our system primarily targets first two class of applications and can work with using camera alone, or commodity depth sensors alone.

2. SYSTEM OVERVIEW & CHALLENGES

This section first presents the overview of the Soft-Swipe system.Then the challenges in enabling V2I (Vehicle-to-Infrastructure)

pairing based on position are described.

2.1 Overview

Soft-Swipe enables position based V2I pairing of vehicles entering into a multi-lane service station. This is performed by matching motion signatures generated from two types of sources. First, Soft-Swipe needs a signature from the vehicle being serviced, and tagged with the vehicle's identity. This signature is received by the infrastructure using V2I links from a device such as a smartphone. The smartphone can fetch the motion profile from vehicle system by using OBD-II port or by wireless links. Next, Soft-Swipe needs signatures for the same vehicle generated by external, *location aimed* devices, that is, devices that are targeted at the locus of interaction, such as a video camera whose field of view covers the multi-lane service station. Note that these signatures are not tagged with the vehicle's identity, because the external devices only know that there is a vehicle in their field of view, but do not know which vehicle it is. Finally, note that multiple sensors may be used to provide complementary or additive information. For external *location-aimed* sensing, cameras, ultra-sonic range sensors, or passive Infrared sensors [12] may be used. In addition, LIDAR, RADAR and microwave technologies that do motion estimation by measuring Doppler shifts can be used as well. Finally, electromagnetic sensing devices such as Inductive coils [28] may be used to detect the presence of metallic bodies, and potentially their velocity.

Figure 1 depicts the architecture of Soft-Swipe where the internal signature is generated by a service device in the vehicle. The external, location-aimed signatures are sensed from two sources (*sensing*): (a) a video camera aimed at the service lane and (b) an array of depth sensors above the service lane and parallel to it. Soft-Swipe uses the two types of signatures in two important ways (detailed in §3.1 and §3.2). First, during system initialization, the motion signatures received from V2I links are used to *calibrate* the external sensing components. This allows these devices to properly convert the phenomena they detect (such as, a series of images, or the distance between where the sensor array is mounted and a planar surface of the automobile) into motion signatures. When the system is in operation, the generated signatures from vision and sensors are combined adaptively (*collaboration*) for obtaining an accurate motion signature. The technique to combine the measurements from the two sensors at a given time instant is described in §3.3. The accurate motion signature thus obtained is sent to a centralized server-side signature matching module. Here, the external motion signatures are matched to the internal motion signature (*matching*) that contains the identity of the object. The mechanism used for matching the motion signatures is described in §3.4. When proper matching occurs, Soft-Swipe can identify the moving object in the sensing field of view, and by definition in the systems proximal locus of interaction.

Soft-Swipe's implementation and cost estimation details are mentioned in §4 and evaluated in §5. Then, §6 contrasts Soft-Swipe with several works in the literature. Finally, §7 presents possible future directions of Soft-Swipe .

2.2 Challenges

Soft-Swipe deals with several practical issues during the sensing, collaboration and matching phases as described below:

- *Uncalibrated sensors:* The camera-based system lacks depth information. Thus, it cannot accurately compute the velocity of the moving object. We need to calibrate the camera for translating the speed as projected onto the camera plane, i.e, pixels per second, to the common unit of meters per second.
- *Rapidly changing speed of vehicles needs fine granular measurements:* For *class-I and II* applications, typical motion signatures include velocities ranging from tens of meters/second (30 to 40 kmph) to a third of a meter per second (1 kmph), with the speeds changing rapidly (such as when a vehicle accelerates away or decelerates to be serviced). Thus all the sensors (both internal as well as external) have to measure a wide range of speeds that may vary rapidly.
- *Differences in sensor capabilities and characteristics across the different types of sources:* Each of the types of sensor components used for motion profile estimation have different error rates, accuracy and precision. Collaboration and matching motion profiles across sources thus cannot be performed simply by adding or matching sample readings, even if they were converted to common units.
- *Noisy environment and multiple vehicles:* Soft-Swipe must be able to distinguish amongst objects in dense environments. For example, in a toll-booth application vehicles are just a few feet apart when they move through the toll booth. Thus, the camera sensor will see a stream of successive vehicles, with potentially more than one vehicle in its field of view, or not see a vehicle, because another vehicle is hiding it from view.

3. SOFT-SWIPE COMPONENTS

This section presents details of the *sensing*, *collaboration* and *matching* algorithms introduced in the previous section. As we mentioned earlier, our implementation uses two types of uncalibrated, commodity sensors to generate the external *location-aimed* motion signatures, namely, a camera or an array of ultrasonic range sensors, and a built-in device (smart-phone) to generate the *object-tagged* internal motion signature.

3.1 Automatic calibration of Camera exploring V2I links

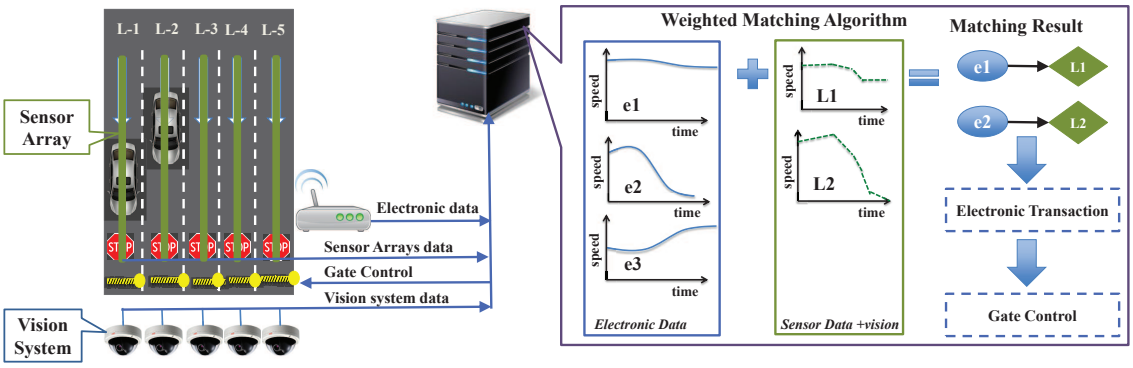

Figure 1: *Soft-Swipe architecture*

This section presents a technique for automatic camera calibration exploiting the V2I links. Since a simple camera does not measure the depth of objects in its field of view, the camera needs to be calibrated in order to convert from the rate at which objects move in the camera plane (which is referred as *optical-flow* [10, 20]), and measured in pixels per second, to the actual velocity of the object being observed. Camera calibration involves finding its height and orientation w.r.t the ground plane. Finding exact orientation and height of the infrastructure cameras is a tedious process. Therefore, prior works have resorted to *automatic camera calibration* techniques which are mainly studied in context of traffic camera installations. Most of these techniques are vanishing point based approaches where they assume lane markers [53, 17], vehicular size [16], road lines [46], straight line motion of vehicles [21] to calibrate a camera. However, in contrast Soft-Swipe explores V2I communication capability of vehicles to calibrate the infrastructure cameras.

Soft-Swipe exploits the *repetitive nature* of vehicles traversing through a station to extract precise scaling values to convert the optical-flow to vehicle's velocity. Our work is similar to the vehicular speed estimation techniques exploiting lane markers [50], where the movement of vehicle across the known length (length of a marker) is used to derive the speed of the vehicle. Instead of assuming known geometries or markings on the ground, Soft-Swipe exploits V2I communication to know how much a vehicle has moved during initial calibrations. Essentially, given two pixel locations L_1 and L_2 along the trajectory of the vehicle, Soft-Swipe obtains distance between these pixel positions on the ground plane from the velocity of the vehicles obtained from V2I link. Similarly, for every pixel Soft-Swipe identifies a scaling value for converting optical flow values to the actual speed on the ground. Vehicles can be detected and tracked across frames using haar based vehicles detectors [44], Deep Neural Network (DNN) [39] based approaches. In our implementation, we have created an *angular filter* exploring V2I communication to mitigate the effects of vehicle detection errors. This angular filter learns the direction of vehicular motion in the camera frame during calibration runs. During the system operation, Soft-Swipe projects the optical flow vectors along this direction. This helps in cases where the vehicles are not detected by the vehicular detectors. The auto-calibration then works as follows:

- After the camera is placed, a single test run is made by the vehicle.
- Soft-Swipe collects the pixel-based location-directed, external motion signature from the camera.
- Soft-Swipe collects the object-tagged motion signature from the device inside the car from V2I link.
- By comparing the two motion signatures, Soft-Swipe calibrates the camera by building a mapping function that translates from pixels/second to meters/second across the path of the moving object. Essentially, the mapping function is a location dependent scaling multiplier that converts from optical speed to actual speed. Note this scaling values is dependent on the pixel position, therefore, a table of scaling values is created.
- Soft-Swipe studies the directions of the vehicular motion in the camera frame and creates angular filters for these directions.

The above technique is used to obtain scaling values and vehicular movement directions. Using these scaling values, for every vehicle entering into the station, Soft-Swipe converts optical-flow to fine granular motion profile. In our implementation we have used a camera with 28 fps (frames per second) which gives a velocity at a granularity of 1/28th of a second. For improving granularity of the motion profile cameras with high fps can be employed.

3.2 Sensor-Fence: Fine-Grained Motion Profiling with Array of Range Sensors

This section presents fine granularity motion profile estimation using an array of commodity depth sensors. The array is hung from the ceiling and is parallel to the ground as shown in Figure 2 and each lane is equipped with one

such sensor array that covers the entire vehicular service station. Inexpensive ultrasonic range sensors [8] that are typically used as robot-eyes [29] are used in our sensor array. Prior work have mainly used ultrasonic sensing and communication to *decipher location information* [57, 45, 37, 34, 19] and *sensing shape* of the objects [13, 35]. In contrast, Soft-Swipe explores *coarse ultrasonic based shape sensing* to first detect a vehicle's existence and then smartly track it for estimating motion profile.

Prior work [56] exploited the roadside sensor deployments to track the vehicle's time of travel between sensors for speed estimation. Let's refer to this approach as *trigger-speed*. As a vehicle enters the lane it triggers each sensor i at a unique time t_i. The average speed between when a vehicle is detected by sensor i and the next sensor $i+1$ is given by $\frac{D}{t_{i+1}-t_i}$, where D is the distance between consecutive sensors (K in numbered). This method generates $K-1$ speed estimates.

By using a different approach, the same sensor array can be used to compute a finer granularity motion profile. Essentially, Soft-Swipe uses closely placed robot-eyes (ultrasonic sensors) to precisely estimate the shape of the vehicle at a given time, and tracks this shape with time across a chain of sensors. As a result, the shape of the vehicles (car, truck etc.) is a by-product which can be used by different toll applications. To begin with, shape estimation is performed by modeling a vehicle's body as a set of planes $\{P_1, P_2, P_3, ...P_n\}$ with a corresponding set of slopes $\{m_1, m_2, m_3, ...m_n\}$. Let us assume that consecutive sensors numbered i and $i+1$ are pointing to the same plane P_j and the vertically traveling signals from these sensors meet the plane at points **A** and **B**, respectively, as shown in Figure 3. The depths observed by these sensors are h_i and h_{i+1}, respectively. Then the slope of plane P_j is estimated as $m_j = (\frac{h_{i+1}-h_i}{D})$. In time Δt, the vehicle moves ahead by $V\Delta t$, and the height reduces by $V\Delta t m_j$. Therefore, the speed of the vehicle at current instance can be estimated by observing rate of change of depth and above computed slope measurement. Let us refer to this approach as *Sensor-fence* since it uses the sensors as a fence to determine the speed of the vehicle. As the sampling rate of these sensors is quite high (20 samples/sec), we can obtain a much finer grained motion profile of the vehicle. For example, for a vehicle moving at 10 mph, with 20 sensors placed at a separation of 2 feet, we can obtain more than 1000 samples in contrast to 20 samples obtained using the *trigger-speed* approach.

To deploy a real system based on the above concept the following practical aspects need to be considered: *(i) Measurement across different planes:* If the points **A** and **B** are on different planes, we cannot use the above technique. For two points on the same plane, their rate of change of depth must be the same, i.e., $(\frac{\Delta h_i}{\Delta t} = \frac{\Delta h_{i+1}}{\Delta t})$. If these rates are not the same, then the sensor reading pair must be discarded. *(ii) Number of sensors:* A larger number of sensors is needed to handle a wide range of speeds. *(iii) Sensor density:* As the sensor density increases the inter-sensor distance decreases. If the sensors are very close, they will see similar heights leading to noisy estimates of the speed. *(iv) Sampling time:* If the sampling rate is very high, the depth difference observed within a sample time will be small and affected by the noise floor. *(v) Noisy Samples:* Some of velocity samples estimated are prone to noise due to the flat shape of a plane on the vehicle. Only if the depth difference $h_{i+1} - h_i \gg 2\sigma$, then the measurement must be used to estimate the speed. The sensor array based system is inexpensive and can work even in dense vehicular environments with a wide range of speeds.

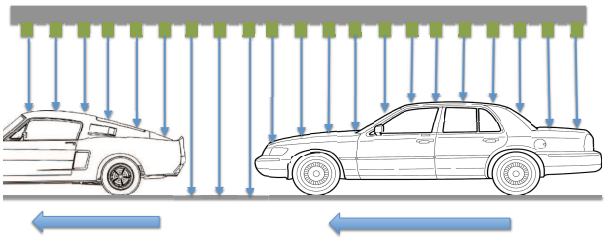

Figure 2: *Sensor fence design. It provides: 1) Highly accurate shape and speed estimation of vehicles; and, 2) Distinguishes very close-by vehicles.*

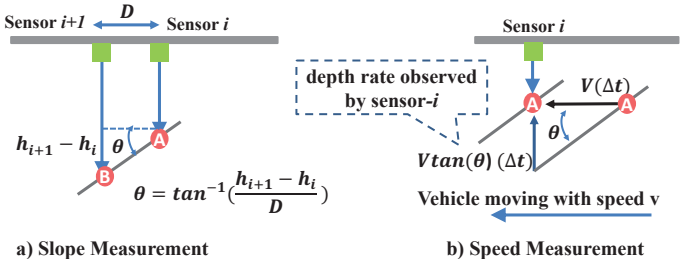

Figure 3: *Speed calibration from sensors: a) Two points A,B on the vehicle close to each-other can be used to measure the slope of the plane. b) Speed of vehicle is measured using this slope and rate of change of depth observed by sensors.*

3.3 Collaboration: Adaptive Weighted (AW) Vision & Sensing

This section attempts to *combine the motion profiles obtained from a camera and sensor-array for obtaining more accurate motion profile*. First, the properties of speed estimation using the sensor-array and vision systems at a given time are studied then, an adaptive weighted scheme for accurate motion profile is designed. In addition, this section automates the calibration and modeling of sensors required for adaptive fusing of motion profiles.

Parameters impacting vision and sensing systems: The experimental data depicts that the vision system performance varies with *Distance from camera*. As the distance between the vehicle and the camera increases, its observability in the

frame decreases and eventually devolves into ambient noise beyond some point. Hence, the speed measurement accuracy decreases with increase in measurement distance. The sensor-array motion profiling performance depends on the *Angle of measurement (θ)*. Soft-Swipe estimates the velocity by measuring the slope of a plane (say, θ). Figure 4 presents the velocity estimation accuracy for planes observed from a vehicle. The slope of these planes are measured by observing depth difference between consecutive sensors which will be affected by the noise floor. Therefore, the slope measurement is not accurate for smaller angles. Notably, accuracy increases with the angle, but the chance of having higher angle planes on vehicle with horizontal spread of inter-sensor distance is low. The best angular plane observed by the sensor array is the windshield.

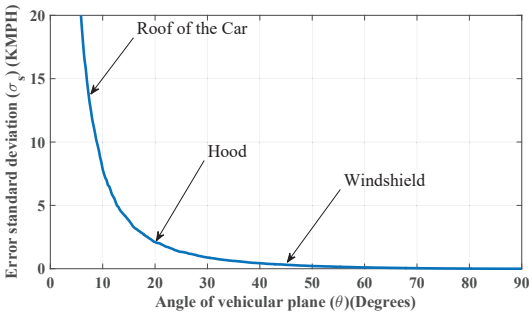

Figure 4: *Simulating sensor-array with different angles. The higher the angle the better the accuracy of slope estimation.*

Combining vision and sensor data: Two major conclusions can be obtained from the previous discussion. First, the accuracies of the sensor-array and the vision system depend on parameters independent of the other system which change with time. Second, these parameters need to be calibrated and studied for accuracy of measurement before using the system.

Prior approaches in sensor fusion fall into two categories; *(1) Dependent sensory measurements*, where multiple sensor measurements are dependent on each other. One example is widely used techniques for fusing data from inertial sensors such as Kalman filter [27] where different observations (such as accelerometer, GPS) are fused by exploring the relationship between these measurements. *(2) Independent sensory measurements*, where different sensors sense for same quantity using independent techniques. One example is EV-Loc [55] where location observations from two sensors (camera and Wi-Fi RSSI) are fused in an adaptive fashion. Similarly, Foresight [32] combines observations from different domains based on distinguishability (or reliability) in each domain. Soft-Swipe belongs to the second category where independent measurements from sensor array and camera are fused. However, in contrast to the above schemes, fusing the motion profiles in the context of Soft-Swipe has additional difficulties due to *(1) Dependency on observable parameters:* Errors are dependent on observable parameters such as distance from the camera and slope of the plane; *(2) Time variant errors:* The measurement errors depend on abovementioned parameters which change with time. Considering these observations, Soft-Swipe first creates an association table of observed parameters and error variance during the training phase. Using this association table, Soft-Swipe combines the vision and sensor motion profiles by *computing the weights for each sample* for accurate fine granular motion profile.

The collaboration between the camera and sensor-array deployed in each lane is enabled by fusing their independent velocity measurements adaptively. Let the velocity measured by camera and sensor arrays be $\hat{v}_c[t]$ and $\hat{v}_s[t]$ respectively at time t in a given lane, then the velocity estimated by combining, $\hat{v}[t]$ will be

$$\hat{v}[t] = w_c[t]\hat{v}_c[t] + w_s[t]\hat{v}_s[t], \tag{1}$$

where $w_c[t]$ and $w_s[t]$ are the weights of camera and sensor array measurements, respectively. These parameters quantify the confidence or accuracy of individual measurements. The camera and sensor measurements can be modeled as $\hat{v}_c[t] = v_r[t]+e_c[t]$ and $\hat{v}_s[t] = v_r[t]+e_s[t]$ where $v_r[t]$ is the real velocity of the vehicle and $e_c[t]$, $e_s[t]$ are measurement errors of the camera and the sensors respectively. Therefore, $E(e_c[t]) = E(e_s[t]) = 0$. Let the variance of $e_c[t]$ and $e_s[t]$ are $\sigma_c^2[t]$ and $\sigma_s^2[t]$, respectively. Also the weights must be normalized, therefore $w_s[t] = 1 - w_c[t]$. The error in combining is $e[t] = w_c[t]e_c[t] + w_s[t]e_s[t]$. Minimum mean square error (MMSE) estimation of velocity reduces to minimizing error variance σ_e^2 as shown below:

$$E(e^2[t]) = \sigma_e^2[t] = w_c[t]^2\sigma_c^2[t] + (1 - w_c[t])^2\sigma_s^2[t]. \tag{2}$$

This mean square error is minimized for

$$w_c[t] = \frac{\sigma_s^2[t]}{\sigma_s^2[t] + \sigma_c^2[t]}. \tag{3}$$

Note the error variances of camera observation $\sigma_c^2[t]$ and sensor observation $\sigma_s^2[t]$ are functions of observable parameters such as angle of plane θ and pixel position $[x, y]$ which are function of time t. In order to estimate $w_c[t]$, abovementioned error variances must be associated with parameters such as slope of plane etc. This involves modeling the sensor array and vision systems and manual calibration for system parameters such as height of camera placement, angle of camera tilt etc. Large sample sets are needed to estimate them accurately. Since modeling the system and observing large sample sets require considerable effort and manual intervention, we instead automate the system using a simple yet intelligent learning and estimation techniques as described below.

Learning Phase: The training set is created and updated in two phases. First during the training phase, for each lane, the user performs trial runs to create different possible $([x, y], \theta)$ pairs and measures $\hat{v}_c[t]$ and $\hat{v}_s[t]$. Along with the estimated

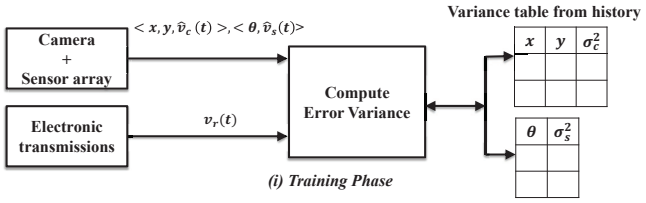

Figure 5: *Figure representing data-flow while estimating weights for MMSE estimation from history table.*

velocities the training set contains associated real velocity v_r, which is obtained from the vehicle's electronic messages. Second during the test phase, if there is only one vehicle in the vehicle-station, then the electronic transmissions of corresponding vehicle is used to train the system deployed in its lane. During this test phase, both vehicle transmissions and sensor observations are added to this set providing a large training set whose size increases with time. Figure 5 presents these two phases and the table construction. With this continuous training set, the sample variances $\sigma_c^2[t], \sigma_s^2[t]$ are incrementally estimated and an association table is created for parameters $([x, y], \sigma_c^2[t]), (\theta, \sigma_s^2[t])$. Also a smoothing function is applied on this table to average close observations creating a continuous trend of variance. Figure 6

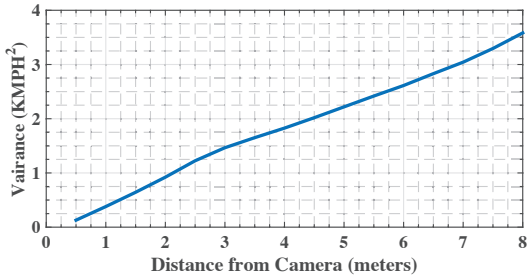

Figure 6: *Camera speed estimation error variance is plotted with vehicle-position from camera frame over 25 experiments.*

presents $\sigma_c^2[t]$ plotted as a function of distance from camera using the history table for 25 experiments. This distance from camera is mapped to pixel-position using a fixed transformation function obtained during training.

Estimating the velocity: Often vehicles traveling in the same lane with similar build (e.g., car, truck etc.) have repetitive (x, y, θ) values. As a result of this for repeating (x, y, θ), the variances can be looked up from the table. From the variance obtained from the look-up tables, the weight $\hat{w}_c[t]$ is estimated using Equation 3 which gives the velocity as $\hat{v}[t] = \hat{w}_c[t]\hat{v}_c[t] + (1 - \hat{w}_c[t])\hat{v}_s[t]$. The estimated velocity $\hat{v}$ at each time t has different measurement errors which must be considered when computing the motion profile of a vehicle over a time-interval. This measurement error is quantified by the variance of measurement $\hat{\sigma}^2[t] = \frac{\hat{\sigma}_c^2[t]\hat{\sigma}_s^2[t]}{\hat{\sigma}_s^2[t]+\hat{\sigma}_c^2[t]}$ which is derived using camera measurement error variance $\hat{\sigma}_c^2[t]$ and sensor measurement error variance $\hat{\sigma}_s^2[t]$ obtained from table look-up using Equation 3.

3.4 *Weighted Matching* of cross domain motion signatures

This section describes the matching component of the system that matches observations obtained from two domains, sensors in each lane and the motion profile from vehicles. In particular, lane observations are matched with electronic messages from vehicles. The accurate motion profile for each lane is obtained using the technique described in the previous section. Similarly, vehicles transmit their motion profile using electronic messages. Essentially if the observations get matched to a vehicle, then it is allowed gate access. If not, Soft-Swipe calls for manual transaction. This section first describes the challenges in cross domain matching followed by presenting a novel metric *Weighted Euclidean Distance* quantifying the closeness between cross domain motion profiles. Using this metric, the rest of the section presents matching and various decisions derived from it.

Challenges in cross-domain matching: As described above, the matching is performed between two domains (sets of data). First, the **electronic identities** (e.g., IP-Addresses or MAC-Addresses of smart-phones) is communicated to Soft-Swipe 's central server over the wireless medium. These electronic identities (e_i) are associated with their motion profile which is received as a packet stream holding velocity and time. Also these electronic motion profiles are assumed to be highly accurate and sampled at a high sampling rate. Second, the **observations** (o_j) from the sensors in each lane are communicated over the wired infrastructure. These observations will be holding the lane-identity and position (position in a lane), current time (t), observed velocity of the vehicle $(v_j^o[t])$, and the accuracy of observation $(\sigma_j^2[t])$. Note that the observed velocity is the adaptive weighted version of vision and sensor array and is output of the algorithm described in previous section.

There are two critical challenges in matching electronic messages with observations. **Different accuracies of measurements:** The speed estimation accuracy obtained from an observation change with time depending on different parameters described in §3.3. If this effect is not considered then noisy observations at one time instant can render the accurate observations at other times useless. **Defective (or) tampered equipment:** There is no guarantee that the vehicles are transmitting their motion profiles. Lack of electronic messages from a vehicle can cause errors in matching.

These two challenges make the problem of matching motion profile distinct from the problems explored in the literature. Traditionally, Euclidean distance [38] and Dynamic time warping (DTW) [54] are methods employed for finding the distance between two time series. But these methods cannot handle the noise or non-uniformity in the measurement errors. Longest Common Subsequence (LCS) is

proposed to handle possible noise that may appear in data; however, it ignores the various time-gaps between similar subsequences, which leads to inaccuracies. Considering this, Soft-Swipe first defines a weighted version of Euclidean distance referred as *Weighted Euclidean Distance* to compute the similarity between two time series which can handle noise. Then, Soft-Swipe uses the above metric to match vehicles with respective observations.

Weighted Euclidean Distance: Non-uniformity in measurement accuracies is addressed by giving weights to the observations based on accuracy. To derive weights based on accuracy (variance of observation), consider an observation o_j with motion profile spanning in a time window $[T_j^o, T]$ containing M_j samples. This motion profile represents a point in M_j dimensional space. Let us define *Weighted Euclidean Distance*($D = \Sigma_{t=T_j^o}^{t=T} w_j[t]^2(\hat{v}_j^o[t] - v_j^o[t])^2$) between two motion profiles as square of distance between two points in the multi-dimensional space, where each dimension is scaled by a weight. These weights ($w_j[t]$) are chosen such that the distance between motion profile of o_j and its accurate measurement ($\hat{v}_j^o[t]$ obtained by electronic messages) must be minimum. In such a case, the distance D is same as mean square error due to measurement noise (discussed in §3.3) and can be formulated as given below

$$E(\Sigma_{t=T_j^o}^{t=T} w_j[t]^2(\hat{v}_j^o[t] - v_j^o[t])^2) = \Sigma_{t=T_j^o}^{t=T} w_j[t]^2\sigma_j^2[t]. \quad (4)$$

Also the weights must be normalized over time. Therefore the objective function D which can be formulated as:,

$$\begin{aligned} \underset{w_j[t]}{\text{minimize}} \quad & \Sigma_{t=T_j^o}^{t=T} w_j^2[t]\sigma_j^2[t] \\ \text{subject to} \quad & \Sigma_{t=T_j^o}^{t=T} w_j[t] = 1. \end{aligned}$$

From Cauchy-Schwarz Inequality,

$$\Sigma_{t=T_j^o}^{t=T} w_j^2[t]\sigma_j^2[t]\Sigma_{t=T_j^o}^{t=T} \frac{1}{\sigma_j^2[t]} \geq (\Sigma_{t=T_j^o}^{t=T} w_j[t])^2 = 1. \quad (5)$$

Therefore,

$$\Sigma_{t=T_j^o}^{t=T} w_j^2[t]\sigma_j^2[t] \geq \frac{1}{\Sigma_{t=T_j^o}^{t=T} \frac{1}{\sigma_j^2[t]}}. \quad (6)$$

The above minimization function is optimized for $w_j[t]\sigma_j^2[t] = K\ \forall t \in [0, T]$ where K is constant.

The optimal weights can be estimated from the variances of each observation as $w_j[t] = \frac{\frac{1}{\sigma_j^2[t]}}{\Sigma_{t=T_j^o}^{t=T} \frac{1}{\sigma_j^2[t]}}$ The computed weights are based on accuracy of measurement as the weight is inversely related to the variance of the observation. Further for a significantly large number of samples, the distribution of D can be approximated as a normal-distribution with mean of $\mu_{Dj} = \frac{1}{\Sigma_{t=0}^{t=T} \frac{1}{\sigma_j^2[t]}}$, with variance of $\sigma_{Dj}^2 = \frac{\Sigma_{t=T_j^o}^{t=T} \sigma_j^2[t]}{\Sigma_{t=T_j^o}^{t=T} \frac{1}{\sigma_j^2[t]}}$. This distribution of D for observation o_j is used to detect corresponding electronic match.

Matching and Fault Detection: Soft-Swipe considers observations that have crossed a threshold length for matching (15 to 20 seconds is found to be optimal in our experiments). With this data, matching happens in a time slotted fashion, and all the observations crossing this threshold in the current time slot are matched in the next time-slot. Also time-slot length is chosen to be much larger than threshold length.

In order to perform matching, the user defines a parameter c (*Match Confidence*) lying between 0 and 1. Matching for an observation o_j, is performed using the abovementioned weights and c. Then Soft-Swipe computes *Weighted Euclidean Distance* $D[i, j]$ for every observation o_j and electronic identity e_i to determine the following:

- If e_i is a correct match for o_j, then the distance $D[i, j]$ is the smallest $\forall i$ and $D[i, j]$ is in high confidence region of normal distribution. (*Match*)
- If o_j has no correct match, then the distances $D[i, j]\forall i$ are not in high confidence region of normal distribution. (*Fault, blocked for manual processing.*)
- If e_i has not matched with any $o_j \forall j$, then e_i is carried over to the next time slot. (*Vehicles yet to enter the station.*)

4. IMPLEMENTATION

In this section we outline our system implemented in vehicular manufacturing and testing station.

Vision system: Our vision system is implemented in C++ using open source computer vision libraries (OpenCV) which captures real-time video feed and finds good features in the frame that can be used to track a vehicle (described by Shi et al. [43]). These features typically include corners, boundaries of a vehicle etc. Once these features are extracted, the vision system checks how these features have moved across consecutive frames in order to measure their shift. These shifts are observed in terms of pixels per unit time and referred as *optical flow vectors* in computer vision literature [10, 20]. The optical flow vectors from different feature points on the vehicle are aggregated to obtain the vehicle's velocity in the camera plane. Next, a noise-filter is created to filter out the optical flow vectors that are below a threshold and not in the directions of vehicular movements. This threshold is determined during the initial calibration runs. Also, the pixels that do not corresponds to any lane can be removed by using image segmentation (segmenting the image corresponding to lane). Small changes in light-conditions, reflections from moving object on the ground and background human movements create optical flow vectors with much smaller magnitudes and in different directions compared to optical flow vectors of a moving vehicle and are filtered out.

The vision system was implemented using a commodity Logitech Quick-cam pro camera and was mounted 2 me-

ters over the ground level. Additionally, we have experimented with Belkin NetCam HD+ and other off-the-shelf digital cameras. The camera must be mounted at a significant height in order to ensure coverage and to approximate a vehicle's motion to a straight line in the camera plane.

The vision system assumes that a vehicle is a solid object and therefore, the system is not trained to look for specific visual features (such as shape of the car, car logo etc.). Feature based vehicle detection and tracking mechanism (where the vehicle can be classified as car, truck etc.) can certainly be layered on Soft-Swipe . Also, the visual-features (described by Li *et al.* [32]) could be used for matching. However, these visual-features cannot distinguish identical vehicles. Soft-Swipe , on the other hand, gives accurate matching without depending on vehicle-specific properties.

Sensor array: The Sensor array is deployed using four ultrasonic sensors [8], which are controlled by Arduino Yun [11] controller as shown in Figure 7. The inter-sensor distance is 30 cm and covers only 90 cm of the vehicle service station. The Sensor array measures the depth at a constant rate of 20 per second and these measurements are processed by Arduino to obtain parameters such as slope or velocity of a vehicle etc. First, the presence of a vehicle is detected by recording the number of sensors triggered at a given time instance. Other triggers (such as caused by a walking person) will usually trigger a small set of sensors and can be ignored. Then the measured velocities along with the parameters are sent to the central server (implemented in a Laptop) using serial communication.

Motion profiles from vehicles are collected by connecting a smart-device with OBD-II system. Adaptive weight and matching components are implemented in Matlab R2015a where the data from vision-system, serial port communication (Arduino) and vehicle smart-device are fetched and processed. The above implementation uses commodity sensors with an average cost of 250 USD per lane. Large-scale production of the system might cost much lower than presented costs.

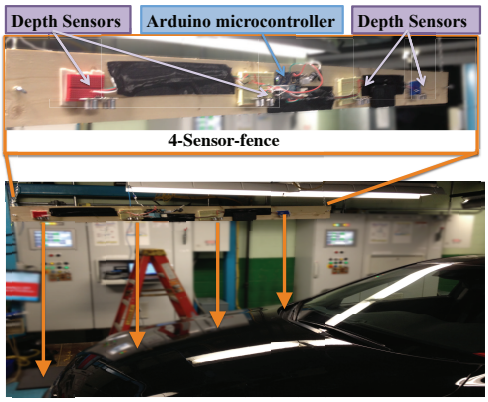

Figure 7: *Sensor fence deployed with four ultrasonic sensors.*

5. EVALUATION

This section evaluates the motion profile accuracy of vision system, sensor array and Adaptive weight algorithm. Then, different metrics for evaluating Soft-Swipe are presented and evaluated with extensive real world experiments.

5.1 Motion-Profile Accuracy

Vision system performance : Our vision system is robust to background noise and estimated speed with an overall standard deviation of 2 kmph and less than 0.5 kmph with a large training set as shown in Figure 8 (i). In evaluating vision-system we observed a variable accuracy achieved in speed sensing. This can be explained as follows, Soft-Swipe calibrates the pixel speed from raw frames and converts this pixel speed to true speed by multiplying with a scaling value. This scaling value is derived for each pixel position during initial training runs. Each training run gives scaling values for a few pixels in the frame. However, during system usage, the closest pixel position with a known scaling value is used in that case.

Sensor-fence performance: We have evaluated the 4-sensor array described in §4 by measuring speed measurement accuracy. Figure 8(b) (blue bars) plots the speed measurement accuracy. We observe that the measurement error increases with the speed of measurement. To analyze the trend we have simulated the sensor system by feeding traces containing dimensions of different vehicles and vehicle mobility traces. Figure 8(b) (red bars) plots the accuracy obtained from simulation. Simulation results showed significant performance for higher velocities. This is due to the higher number of sensors needed for capturing higher velocities. The sensor-fence performance depends mainly on the angle of plane as described in *sensor fence* section. But with limited number of sensors (in experiments 4 were used), the chance of capturing higher-slope planes is less as compared to a long chain of sensors (in simulations).

Adaptive weight algorithm performance: We evaluate the benefits of combining the motion profiles obtained from the vision and sensor systems by using the adaptive weight algorithm. Figure 9 plots the motion profile using vision, sensor array and adaptive weight algorithm. The adaptive weight algorithm produces less noisy and more accurate motion profile by combining information from both the vision and the sensor array components. We have also experimented with several naive smoothing algorithms to reduce noise in the process of combining information. But these algorithms miss the sharp peaks in the motion profile (sudden stops, acceleration etc) and therefore are not suitable for dynamic vehicular speeds. For a set of 30 experiments, Adaptive weight algorithm reduced error by 50% (i.e., nearly 1kmph) compared to vision system and 55% (i.e., nearly 1.2kmph) as shown in the Figure 8(c).

5.2 System Performance

This section first presents the metrics involved in evalu-

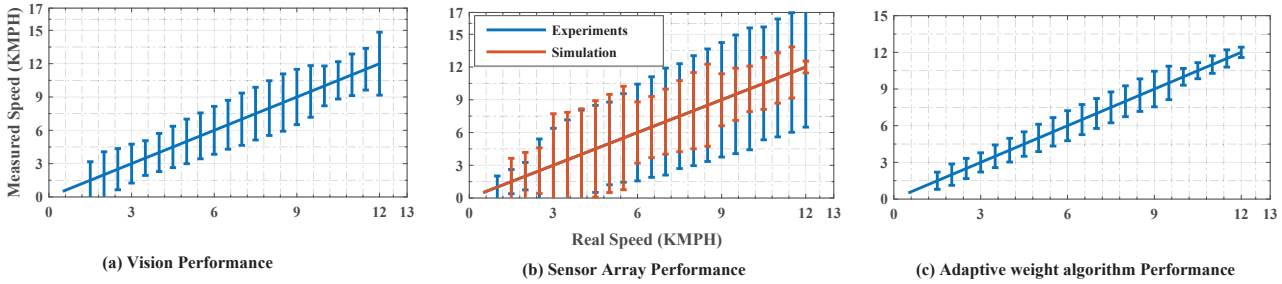

Figure 8: *Speed estimation variance-plots of vision-system (experiments) with average standard deviation 1.6 kmph, sensor-system (simulation and experiments) with average variance of 2 kmph, and adaptive algorithm with average variance of 1 kmph from Indoor low speed experiments. The Adaptive weight algorithm combines sensor simulated results and vision experimental results for estimating the motion profile and reduces the error by more than 50 %.*

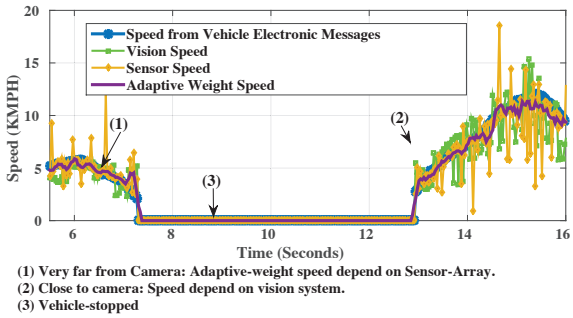

Figure 9: *Motion profile from vehicular Electronic messages, sensor-system, vision-system, and Adaptive-weight algorithm.*

ating Soft-Swipe system. Then, the experimental setup for evaluation is presented, followed by a discussion on results and observations.

Evaluation metrics: To examine the benefits of the matching algorithm, we evaluated the system for following metrics. *(a) Precision (p), Recall (r) and F-Score (f)*: Precision gives the ratio of number of correct matches to the total number of matches produced by the algorithm. Recall gives the ratio of number of correct matches produced by the algorithm to the total number of correct matches (ground truth). F-score (F_1-score) is the commonly used statistical metric quantifying accuracy of matching, considering both p and r. Precision, recall and F-Score are standard metrics defined for matching [36]. In addition, we define the following metrics from the *users point of view* which are important for different toll based applications. *(b) Identity-Swap*: The probability of swapping identity between vehicles. Its the ratio of false-positives to the total number of times an observation (user) participates in the matching. Note this is always less than $1-p$, as $1-p$ is the ratio of false-positives to total number of times an observation is matched. This metric quantifies the probability that a user pays someone else's toll and still got the gate access. This metric is essential for drive-thru and other service based transactions as this metric quantifies the incidence of swapped transactions. *(c) False-stop*: The probability of having a wrong match or no match for a given observation. This includes observations that are considered to have a wrong match (false negatives) as well as no matches and is therefore always greater than $1-r$. *(d) Miss-Rate*: It is the probability of detecting an observation without electronic transmissions (rogue-vehicle). This metric quantifies the probability of having gate access without performing electronic pairing and therefore, is essential for toll based applications.

Experimental setup: First a huge number of single lane experiments are conducted with controlled variation of traffic pattern, just like typical class I and class II applications. Note these applications can often have multiple lanes for reducing wait times. Since building the system for multiple lanes experimental setup is cumbersome, we have designed an emulator which simply replays different or same experiments across different emulated lanes. Therefore, vehicles across different lanes can have same motion profiles. Then multi-lane experiments are created with varying lane-count ranging from 1 to 5. Additionally the system receives motion profiles from 7 exterior electronic transmissions (vehicles yet to enter the station but transmitting the motion-profile). For all the experiments the user defined parameter c is set to 0.99. For evaluating the miss rate, out of the vehicles in the station, one vehicle is assumed rogue, which does not transmit the motion profile. Then the system is evaluated for detecting this rogue-vehicle.

Results and observations: Figure 10 depicts the results observed from the abovementioned experiments. From these results, we observe the following general trends: *Precision increased with number of lanes and swap-rate decreased with lanes*. This trend in precision is mainly attributed to reduction in noise (noise-vehicle transmissions) per lane. Increase in precision rate also results in lower swapping rates. *Recall decreased with number of lanes and False-stops increased linearly with number of lanes*. With more number of lanes, the fraction of noise-vehicles (vehicles yet to

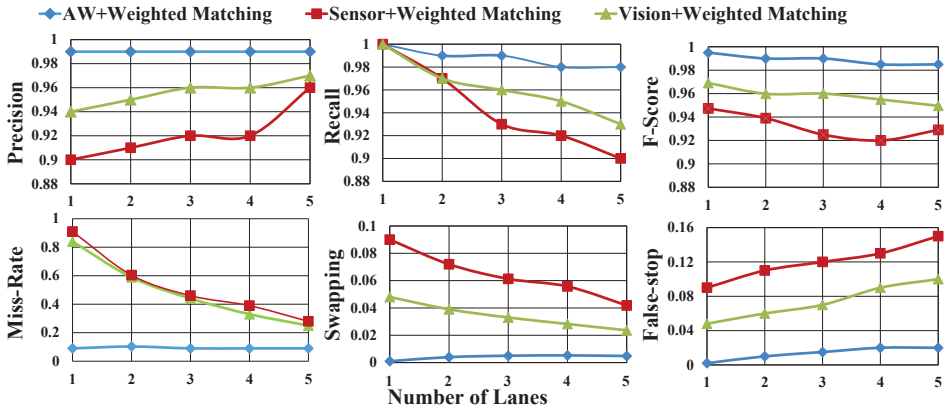

Figure 10: *Weighted matching algorithm is evaluated for different metrics using vision-only, sensor-fence, and Adaptive weight(AW) algorithm.*

enter station) reduces leading more vehicles considered as match. Increase in Recall reduces the precision. When recall is high, the lower precision will result in some vehicles to be stopped for traditional processing (perhaps with manual intervention). We observed miss-rate can be reduced further by increasing the confidence (c) defined in §3.4, but this will reduce the recall leading to valid pairs being eliminated as a miss (rogue-vehicle). This implies the lower the miss-rate, higher the chance of valid vehicles being considered as a miss (rogue-vehicle). Also, by reducing the c, *recall* can be increased, but this reduces the precision.

6. RELATED WORK

Soft-Swipe enables accurate pairing between a vehicle and the infrastructure by exploiting motion signatures of the vehicle at a particular location. Our work is primarily related to following three lines of research.

(i) Motion signatures for identification: [51] exploits visual and discrete motion sequences for identifying the human visually. These position sequences cannot be used to distinguish vehicles as all the vehicles move in the same direction and may have identical visual features. [32] uses position and color to identify vehicles and enable unicast. But position from GPS cannot resolve the vehicle to its respective lane. Also, multiple vehicles can have the same color (e.g., very common in automobile manufacturing plants). [48, 47] uses motion signatures of vehicles observed by a vehicle using its on board sensors such as camera and RADAR to identify neighboring vehicles. In order to enable pairing between vehicles, distinguishable signatures must be extracted with high accuracy, which cannot be achieved by above works.

(ii) Location signatures: Location based signatures are widely explored in the context of NFC, wireless localization and wireless security. The ambient sensors available on the mobile phones such as audio, light, GPS, Wi-Fi, Bluetooth, and thermal are used to create location specific signatures to authenticate [25, 24, 49, 33]. [52] defines motion-signatures, which can be captured by inertial-sensors on mobile phones to provide indoor localization service. [22] have presented techniques to track the user exploring the motion signatures. [15] explores Wi-Fi and Bluetooth RSSI signatures to sense the context of a user. However, Wi-Fi based signatures are heavily time varying in dynamic environments and difficult to sense.

(iii) Vehicle speed sensing and Matching: Prior works have explored road-side camera [23, 41], Soft-Swipe uses a novel algorithm for dynamic speed estimation of a vehicle using both vision and depth-sensor array. Speed estimation algorithm from vision proposed in Soft-Swipe is similar to works on speed estimation from road-side cameras [23, 41]. Soft-Swipe first estimates shape of a moving vehicle using depth sensor array hung from the ceiling. Then, movement of this object across sensor-array length is used to estimate the vehicle speed. The problem of estimating the shape of a vehicle has similarities with the problem of *object construction from 3D points* [40], but Soft-Swipe exploits the 2-Dimensional nature of the speed estimation problem and includes a novel lightweight algorithm for shape and speed estimations.

(iv) Sensor fusion: Prior works [55, 32] have explored the weight adaptation algorithms by using variances of observations. However, we showed that these variances do not remain constant in the context of vehicular speed sensing based applications. Realizing this non-uniformity in the variances, we have proposed a learning based Adaptive weight algorithm to combine motion signatures from multiple modalities by *computing weights for each sample*.

7. FUTURE WORK

The following three ideas related to motion-signatures for enabling general pairing mechanisms in the context of vehicular communications are worth exploring.

(i) Enhancing motion signatures for intra-vehicular pair-

ing: Soft-Swipe exploits motion signatures to securely pair vehicles with the infrastructure. This idea can be extended for pairing intra-vehicular systems in smart-vehicles. Intra-vehicle systems include multiple mobile phones, tablets, navigation system, cruise control, heating etc. These systems can continuously observe the motion profile, which can be used as a secret key to pair these systems. **(ii) Enhancing motion signatures with vehicle localization:** To increase the accuracy of matching, other coarse localization technique such as based on RSSI of the RF signal can be used. The RSSI can be used to limit the set of vehicles to match a particular lane, leading to higher accuracy of matching. **(iii) Enhancing motion signatures with Tagging Infrastructure:** The infrastructure can be tagged or planted efficiently to encode lane specific information. One simple mechanism to encode lane identity is by using minor obstacles such as bumps and potholes. This information can be observed by vehicles inertial sensors and can be used to identify a lane and its corresponding position. Information encoding can be performed by using constructions such as left-obstacle, right-obstacle, complete-obstacle etc. and using multiple such obstacle.

8. ACKNOWLEDGMENTS

This work was partially supported by a grant from Honda R&D Americas.

9. REFERENCES

[1] Apple pay. https://www.apple.com/apple-pay/ (March 24, 2016), 2016.

[2] EZ-pass. https://www.e-zpassny.com/ (March 24, 2016), 2016.

[3] EZ Pass mount error. http://wtop.com/ticketbuster/2015/07/e-zpass-mounting-error-can-cost-hundreds/ (March 24, 2016), 2016.

[4] EZ-Pass unhappy customer report. http://wtop.com/news/2014/04/this-e-zpass-error-can-cost-you-thousands-of-dollars/ (March 24, 2016), 2016.

[5] Fast-Track. https://www.bayareafastrak.org/ (March 24, 2016), 2016.

[6] Google wallet. https://www.google.com/wallet/ (March 24, 2016), 2016.

[7] I-PASS. http://www.illinoistollway.com/tolls-and-i-pass (March 24, 2016), 2016.

[8] Ultrasonic Ranging Module HC - SR04. http://www.micropik.com/PDF/HCSR04.pdf (March 24, 2016), 2016.

[9] VISA Mobile application platform. https://developer.visa.com/paywavemobile (March 24, 2016), 2016.

[10] Kelson RT Aires, Andre M Santana, and Adelardo AD Medeiros. Optical flow using color information: preliminary results. In *Proceedings of the ACM SAC 2008*, pages 1607–1611. ACM, 2008.

[11] Arduino. Arduino yun board. https://www.arduino.cc/en/Main/ArduinoBoardYun (March 24, 2016), 2016.

[12] Abhishek Arora, Rajiv Ramnath, Emre Ertin, Pradeep Sinha, Sandip Bapat, Vinayak Naik, Vinod Kulathumani, Hongwei Zhang, Hui Cao, Mukundan Sridharan, et al. Exscal: Elements of an extreme scale wireless sensor network. In *Embedded and Real-Time Computing Systems and Applications, 2005. Proceedings. 11th IEEE International Conference on*, pages 102–108. IEEE, 2005.

[13] Mitsuru Baba, Kozo Ohtani, and Syunya Komatsu. 3d shape recognition system by ultrasonic sensor array and genetic algorithms. In *Proc of IEEE IMTC 2004*, volume 3, pages 1948–1952. IEEE, 2004.

[14] Dirk Balfanz, Philippe Golle, and Jessica Staddon. Proactive data sharing to enhance privacy in ubicomp environments. In *Proc of UbiComp 2004 Privacy Workshop*, 2004.

[15] Xuan Bao, Bin Liu, Bo Tang, Bing Hu, Deguang Kong, and Hongxia Jin. Pinplace: associate semantic meanings with indoor locations without active fingerprinting. In *Proc of ACM UbiComp 2015*, pages 921–925. ACM, 2015.

[16] Daniel J Dailey, Fritz W Cathey, and Suree Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):98–107, 2000.

[17] Douglas N Dawson and Stanley T Birchfield. An energy minimization approach to automatic traffic camera calibration. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1095–1108, 2013.

[18] Thomas P Diakos, Johann A Briffa, Tim WC Brown, and Stephan Wesemeyer. Eavesdropping near-field contactless payments: a quantitative analysis. *The Journal of Engineering*, 1(1), 2013.

[19] Esko Dijk, K van Berkel, Ronald Aarts, and E van Loenen. Single base-station 3d positioning method using ultrasonic reflections. In *Proc of ACM UbiComp 2003*, pages 199–200, 2003.

[20] Sedat Doğan, Mahir Serhan Temiz, and Sıtkı Külür. Real time speed estimation of moving vehicles from side view images from an uncalibrated video camera. *Sensors*, 10(5):4805–4824, 2010.

[21] Markéta Dubská, Adam Herout, Roman Juránek, and Jakub Sochor. Fully automatic roadside camera calibration for traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1162–1171, 2015.

[22] Xianyi Gao, Bernhard Firner, Shridatt Sugrim, Victor Kaiser-Pendergrast, Yulong Yang, and Janne Lindqvist. Elastic pathing: Your speed is enough to track you. In *Proc of ACM UbiComp 2014*, pages 975–986. ACM, 2014.

[23] Lazaros Grammatikopoulos, George Karras, and Elli Petsa. Automatic estimation of vehicle speed from uncalibrated video sequences. In *Proceedings of International Symposium on Modern Technologies, Education and Profeesional Practice in Geodesy and Related Fields*, pages 332–338, 2005.

[24] Tzipora Halevi, Haoyu Li, Di Ma, Nitesh Saxena, Jonathan Voris, and Tuo Xiang. Context-aware defenses to rfid unauthorized reading and relay attacks. 2013.

[25] Tzipora Halevi, Di Ma, Nitesh Saxena, and Tuo Xiang. Secure proximity detection for nfc devices based on ambient sensor data. In *Computer Security–ESORICS 2012*, pages 379–396. Springer, 2012.

[26] Kiran Joshi, Steven Hong, and Sachin Katti. Pinpoint: Localizing interfering radios. In *Proceedings of the USENIX NSDI 13*, pages 241–253, Lombard, IL, 2013. USENIX.

[27] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[28] Yong-Kul Ki and Doo-Kwon Baik. Model for accurate speed measurement using double-loop detectors. *Vehicular Technology, IEEE Transactions on*, 55(4):1094–1101, 2006.

[29] P Kleinschmidt and V Magori. Ultrasonic robotic-sensors for exact short range distance measurement and object identification. In *IEEE 1985 Ultrasonics Symposium*, pages 457–462. IEEE, 1985.

[30] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. Accurate indoor localization with zero start-up cost. In *Proceedings of the ACM MOBICOM*, pages 483–494, 2014.

[31] Dong Li, Tarun Bansal, Zhixue Lu, and Prasun Sinha. MARVEL: Multiple antenna based relative vehicle localizer. In *In Proceedings of ACM MOBICOM 2012*, pages 245–256, New York, NY, USA, 2012. ACM.

[32] Dong Li, Zhixue Lu, Tarun Bansal, Erik Schilling, and Prasun Sinha. Foresight: Mapping vehicles in visual domain and electronic domain. In *Proc. of IEEE INFOCOM*, pages 1995–2003, 2014.

[33] Di Ma, Nitesh Saxena, Tuo Xiang, and Yan Zhu. Location-aware and

safer cards: Enhancing rfid security and privacy via location sensing. *Dependable and Secure Computing, IEEE Transactions on*, 10(2):57–69, 2013.

[34] Masateru Minami, Yasuhiro Fukuju, Kazuki Hirasawa, Shigeaki Yokoyama, Moriyuki Mizumachi, Hiroyuki Morikawa, and Tomonori Aoyama. Dolphin: a practical approach for implementing a fully distributed indoor ultrasonic positioning system. In *Proc of UbiComp 2004*, pages 347–365. Springer, 2004.

[35] Kozo Ohtani and Mitsuru Baba. *Shape Recognition and Position Measurement of an Object Using an Ultrasonic Sensor Array*. INTECH Open Access Publisher, 2012.

[36] Tan Pang-Ning, Michael Steinbach, Vipin Kumar, et al. Introduction to data mining. In *Library of Congress*, page 74, 2006.

[37] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proc of ACM MOBICOM*, pages 32–43. ACM, 2000.

[38] Davood Rafiei and Alberto Mendelzon. Similarity-based queries for time series data. In *Proceedings of the ACM SIGMOD Record*, volume 26, pages 13–25. ACM, 1997.

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[40] Ruwen Schnabel, Raoul Wessel, Roland Wahl, and Reinhard Klein. Shape recognition in 3d point-clouds. In *Proc. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, volume 2. Citeseer, 2008.

[41] Todd N Schoepflin and Daniel J Dailey. Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. *Intelligent Transportation Systems, IEEE Transactions on*, 4(2):90–98, 2003.

[42] Souvik Sen, Božidar Radunovic, Romit Roy Choudhury, and Tom Minka. Spot localization using phy layer information. In *Proc of ACM MobiSys*, 2012.

[43] Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[44] Sayanan Sivaraman and Mohan Manubhai Trivedi. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1773–1795, 2013.

[45] Adam Smith, Hari Balakrishnan, Michel Goraczko, and Nissanka Priyantha. Tracking moving devices with the cricket location system. In *Proc of ACM MobiSys 2004*, pages 190–202. ACM, 2004.

[46] K-T Song and J-C Tai. Dynamic calibration of pan&# 8211; tilt&# 8211; zoom cameras for traffic monitoring. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(5):1091–1103, 2006.

[47] Gopi Krishna Tummala, Dong Li, and Prasun Sinha. Roadmap: Mapping vehicles to ip addresses using motion signatures. In *Proceedings of ACM CarSys 2016*. ACM.

[48] Gopi Krishna Tummala, Dong Li, and Prasun Sinha. Roadview: Live view of on-road vehicular information. In *Proceedings of IEEE SECON 2017*. IEEE.

[49] Pascal Urien and Selwyn Piramuthu. Identity-based authentication to address relay attacks in temperature sensor-enabled smartcards. In *Smart Objects, Systems and Technologies (SmartSysTech), Proceedings of 2013 European Conference on*, pages 1–7. VDE, 2013.

[50] US Department of Transportation, Federal Highway Administration. *Manual on Uniform Traffic Control Devices*. 2009.

[51] He Wang, Xuan Bao, Romit Roy Choudhury, and Srihari Nelakuditi. Visually fingerprinting humans without face recognition. In *Proceedings of the ACM MobiSys*, pages 345–358, 2015.

[52] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. No need to war-drive: unsupervised indoor localization. In *Proc. of ACM MobiSys*, pages 197–210, 2012.

[53] Kunfeng Wang, Hua Huang, Yuantao Li, and Fei-Yue Wang. Research on lane-marking line based camera calibration. In *Vehicular Electronics and Safety, 2007. ICVES. IEEE International Conference on*, pages 1–6. IEEE, 2007.

[54] B-K Yi, HV Jagadish, and Christos Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Data Engineering, 1998. Proceedings., 14th International Conference on*, pages 201–208. IEEE, 1998.

[55] Boying Zhang, Jin Teng, Junda Zhu, Xinfeng Li, Dong Xuan, and Yuan F Zheng. EV-Loc: Integrating electronic and visual signals for accurate localization. In *Proc. of ACM MOBIHOC*, pages 25–34. ACM, 2012.

[56] Zusheng Zhang, Tiezhu Zhao, and Huaqiang Yuan. A vehicle speed estimation algorithm based on wireless amr sensors. In *Big Data Computing and Communications*, pages 167–176. Springer, 2015.

[57] Yi Zhao, Anthony LaMarca, and Joshua R Smith. A battery-free object localization and motion sensing platform. In *Proc of ACM UbiComp*, pages 255–259. ACM, 2014.